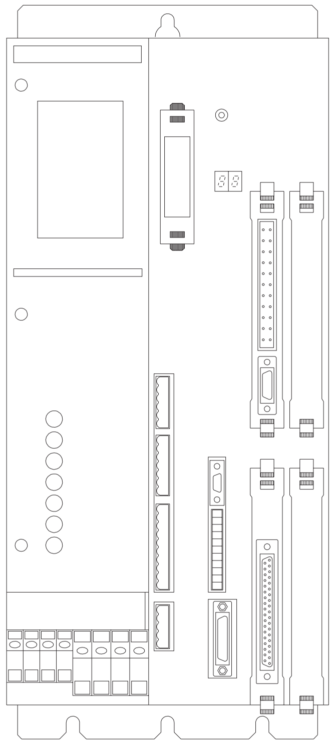


DLC Single Axis Control Card for Firmware Versions DA and DG

User Manual

DOK-CONTRL-DLC*A+DLC*G-AW03-AE-P

Title DLC Single Axis Control Card for Firmware Versions DA and DG

Type of Documentation User Manual

Document Typecode DOK-CONTRL-DLC*A+DLC*G-AW03-AE-P • 12/99

Internal File Reference
- DLC-A/DLC-G User Manual
- 120-0400-B324-03/AE
- Part No. 286834

Purpose of Documentation This documentation is written for both operating personnel and the machine builder. It explains how to interface, install, set up and operate the Indramat DLC Positioning Control with DA or DG software.

Record of Revisions

Description	Release Date	Notes
Initial Release	6/94	Rev. A
Updated/added Parameters Updated Commands Updated/added Status Codes Updated/rearranged Error Codes Added Alphabetical and Numerical Tables of Error Codes. Deleted Appendix C, DLC Program Commands/Format. Deleted Appendix D, Parameter Input Sheets	7/95	Rev. B
Combine DLC-A (Pub. # 68000) and DLC-G (Pub. # 68006) Manuals Parameter additions and changes	12/99	Rev. C

Published by Mannesmann Rexroth Indramat GmbH

INDRAMAT Hoffman Estates • 5150 Prairie Stone Parkway • Hoffman Estates, IL 60192 • USA

Phone: 847-645-3600 • Fax: 847-645-6201

http://www.rexroth.com/indramat

Dept. ESV (HK).

Table of Contents

Rexroth Indramat

1 General Description

The DLC is a single-axis positioning control card which plugs into the Indramat Digital AC Servo Drives, such as the DKS (Digital Compact Drive) or DDS (Digital Drive) products. The DKS is a self-contained power supply and digital drive. The DDS is a digital drive which operates with the TVD (AC Power Supply). The DLC requires a DEA 4 Input/Output card or available I/O network cards to provide the system inputs and outputs to operate the Digital AC Servo System. The DLC plugs into the U1 slot and the I/O card plugs into the U2 slot of the Indramat Digital Drive Modules. The DLC/DEA 4 used with an Indramat Digital Drive Module becomes a multi-tasking, user programmable unit which provides precise motion control. The DLC controls an Indramat maintenance-free MDD Digital AC Servo Motor to drive a ballscrew or some other positioning device. This is a closed-loop feedback system which provides precise control of speed and position at all times. The DLC Digital AC Servo System is used for a variety of positioning applications. The DLC/DEA 4 installed in the Indramat DKS, Digital Compact Drive, is illustrated in Figure 1.1.

Typical applications include:

- Rotary tables
- Packaging machines
- Thermoforming machines
- Gantry robots
- Handling equipment
- Wood working machines

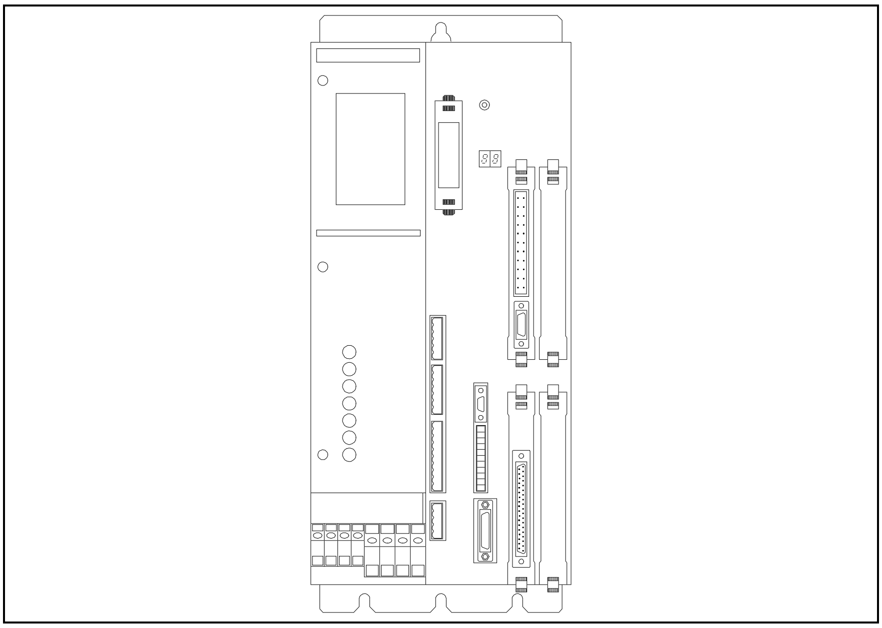


Figure 1-1: DKS Positioning Control Module

The extensive program command set permits the DLC to perform even complex processing tasks. It can do multi-tasking, operating one motion program and two background programs simultaneously. The DLC can be programmed both on-line, and off-line.

The DLC can be used in remote operation, where it is controlled by the customer's line control, usually a computer or a programmable controller, which controls operation of the whole machine. The function of the line control is to convey commands and to receive information via I/O connections from the DLC Digital AC Servo System.

The DLC requires an I/O card be plugged into the U2 slot of the DKS. The DEA 4 card provides 15 inputs (8 system/ 7 auxiliary) and 16 outputs (5 system/11 auxiliary). Optional cards can increase the auxiliary inputs to 37 and the auxiliary outputs to 43. In many applications, the DLC/DEA 4 with an Indramat Digital Controller Module can provide sufficient machine control without the use of an external line control. Other information, such as programs, parameters, and system status can be communicated (two way) between the DLC and a host device, such as a computer, programmable controller or Indramat SOT, via a multi-format serial communications port.

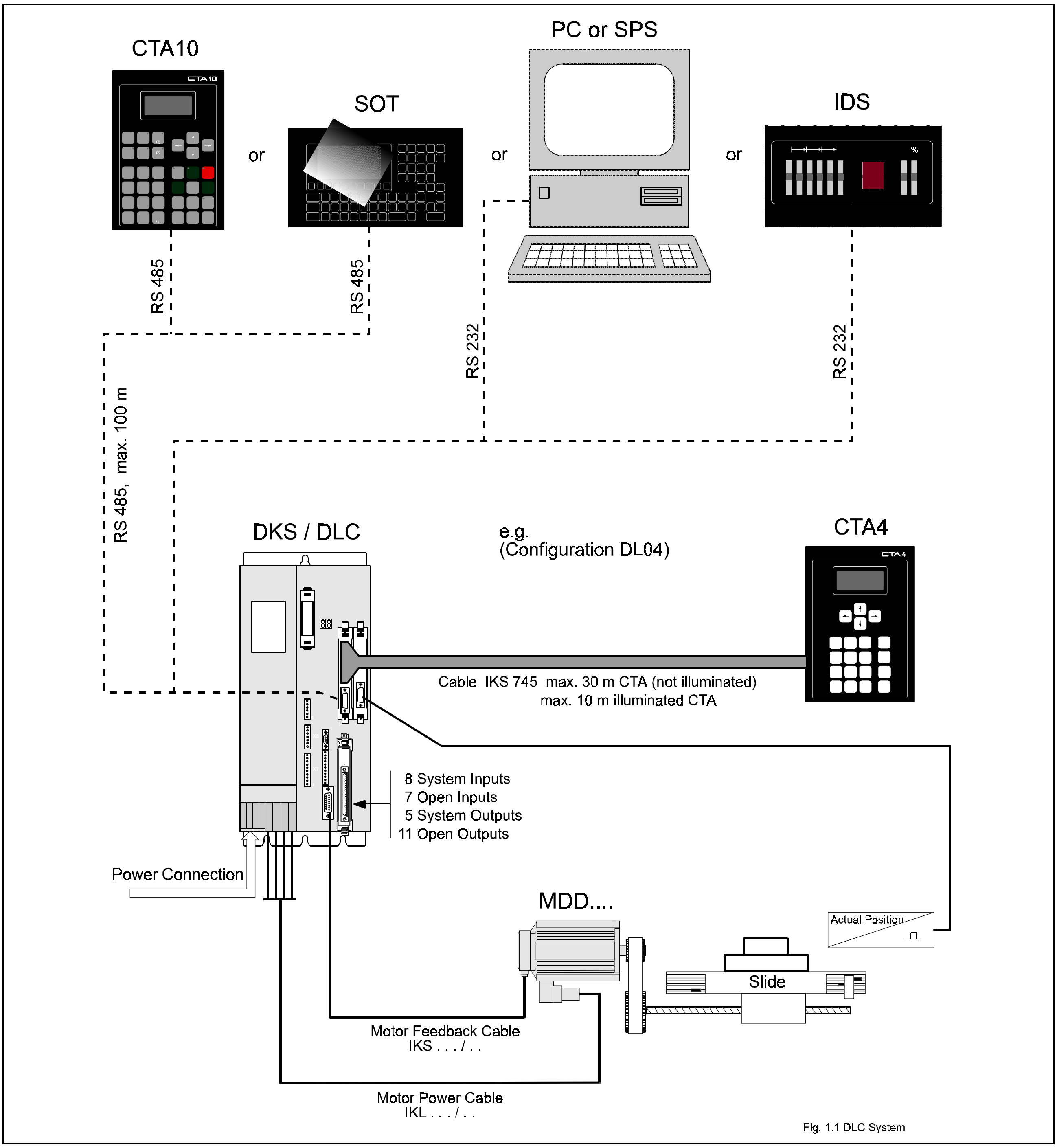


Figure 1-2: Block Diagram

1.1 DLC Configuration

The DLC Digital AC Servo System typical configuration consists of:

- DLC 1.1 control card
- DEA 4.1 Input/Output card
- Digital Drive (DKS/DDS)
- MDD Digital AC Servo Motor

Complete interconnect cable sets are also available from Indramat. The components are chosen to best fit the required application. Figure 1-2 is a block diagram of a typical DLC Servo System configuration. These components are designed into a mechanical system. For example, it could feed some type of material into another processing station, such as a punch press, thermoforming station, packaging machine, etc.

The machine builder or user enters data into the DLC parameters to specify the mechanical and operating characteristics of the system. Based on this data, plus the feed length and feed rate entered by the operator, the DLC issues positioning commands to the digital drive, a DKS (Digital Compact Drive) or DDS (Digital Drive), which controls the current driving the MDD AC Servo Motor, which drives the mechanical feed mechanism.

The MDD AC Servo Motor includes a high-resolution feedback, which provides velocity and position feedback to the digital controller, ensuring precise, repeatable positioning of the material being fed. The final accuracy of the feed system depends on various factors, such as type of material, gearbox backlash and other machine mechanics.

System components are modular, thus installation and replacement of any component of the system is fast and easy. The Indramat Digital Drive Modules and the MDD AC Servo Motor have quick-connect cabling. The drive and MDD Servo Motor are matched for optimum operation using a plug-in DSM module. Thus, should a failure occur, replacement of the digital drive is accomplished quickly without the need for electronic fine tuning. This results in a minimum of lost production because of machine downtime.

The system is designed to ensure operating integrity and safety, using various inputs and outputs for handshaking to assure that the feeder and subsequent processing station or device operate in harmony. A complete diagnostic system monitors all inputs, outputs and operating conditions and stops the system if a fault is detected. Diagnostic messages are displayed to aid the operator in troubleshooting problems and quickly getting the system back into production.

1.2 About This Manual

This document is written for both operating personnel and the machine builder. It explains how to interface, install, setup and operate the Indramat DLC Positioning Control with DA or DG software.

Hardware and Software Support

This manual describes the DLC 1.1 hardware, used with:

- DA software versions DA 01.1-01.6 up through version DA 01.1-03.xx
- DG software versions DG 01.1-02.7 through DG 01.1-03.xx

Indramat provides assistance for any problems you may encounter with this system. Your first source of information should be this manual. To report a problem or request assistance, call Indramat at (847) 645-3600, between 9:00 AM and 5:00 PM Central time. Ask for a Service Engineer or call our 24 hour Service Hotline at 1-800-860-1055. You may also write or FAX to the following:

Rexroth Indramat

Attn: Service Department

5150 Prairie Stone Parkway

Hoffman Estates, IL 60192

FAX: (847) 645-6201

How To Use This Manual

The manual is organized such that Chapters 1 and 2 describe the DLC control and its operation. These chapters, plus Chapter 8 on diagnostics, will be sufficient for most operating personnel. Chapters 3-8 provide functional description, installation, setup, parameter entry, programming, and diagnostic and troubleshooting information required by the machine builder and setup personnel.

Chapter 1 **General Description**

This section describes the DLC control and the features which make it well suited for motion control. Describes and illustrates various options. Lists specifications.

Chapter 2 **Controls & Indicators**

This section describes the CTA keypad and displays interfaced with the DLC control card

Chapter 3 **Functional Description**

This section describes all pre-defined, plus several user definable, input and output signals and the various interfacing and operating modes of the DLC. This information is necessary for interfacing the DLC to the machine builder's equipment, control panel design and troubleshooting.

Chapter 4 **Parameters**

This section describes all user-entered parameters required to adapt the DLC to the mechanical and electrical characteristics of each application.

Chapter 5 **Programming**

This section describes all program commands provided in the DLC for the user to create the executable program, as desired for the application.

Chapter 6 **Installation/Start-up**

This section describes procedures for installing a DLC control system. Provides an example of a DLC start-up and testing procedure.

Chapter 7 **Serial Interface**

This section describes the multi-format RS-232/485 port and the protocol for two way communication between the DLC and a host device.

Chapter 8 **Diagnostics & Troubleshooting**

This section describes the DLC's self-diagnostic system, lists and explains all diagnostic messages and describes troubleshooting procedures.

Appendices

Appendix A **DA and DG Programming Notes**

This section is periodically updated with hints and examples of use for programming commands.

Appendix B **Display Map**

This section shows the DLC display screens which appear on the CTA control panel.

Appendix C **Interconnect Drawings**

This section contains interconnect drawings for

- DKS-DLC
- RS232 Data Interface
- SOT-DLC RS485
- CTA-DLC

A timing diagram for the DLC with DKS,DDS or DDC is also included.

Appendix D **Installation Drawings**

This section contains a DKS 1.1 dimensional outline drawing as well as CTA and IDS cabinet cutout dimensions for remote mounting.

Appendix E **DLC Type Code Descriptions**

This section shows how to interpret the data plate for hardware/software options included.

1.3 System Features

Superior Performance

The system offers high precision motion control with feed resolution of 0.001 inch. Note that maximum system performance depends on the mechanical characteristics of the user's system.

Easy to Operate

The user simply and easily operates the control system by entering a simple user program using the optional interfaces. Operating status messages appear on the display in the user selected language - English, French, German, Spanish, Italian or Portuguese. Other input and display options are described later in this section. The DLC system includes features to make setup quick and easy, eliminating time consuming mechanical setup or complex programming when changing parts.

Parameter-Adaptable to Multiple Machines

The machine manufacturer or the user easily adapts the DLC to the mechanical and electrical characteristics of an application by entering data into a set of parameters, using the DLC's optional CTA keypad and display. These parameters define the characteristics of the machine, such as: maximum and minimum feed lengths, jog, acceleration and deceleration rates, units of feed measurement, RS-232/485 serial communication characteristics, etc. This allows one single type of DLC control to handle the mechanics of various types of different machines. Thus, plant personnel need be familiar with only one control system.

Generally, parameters are entered once when setting up the system, then changed only if the configuration changes or if different types of operations are required. The factory installed DLC executive program interprets the parameters to match the DLC Digital AC Servo System to the machine, and translates operator-entered commands into motion control signals, coordinating the feed motion with the parts of the other machinery. Complicated system programming is not required.

Fully Self-Diagnostic

System protection is paramount. The DLC detects normal operating status, operator errors, errors in the control itself and machine faults.

Both fault and normal status messages can displayed on the DLC's optional CTA interface, in the user selected language. Thus, the operator is informed of the current operating status of the system and is alerted to any condition that causes a fault. This helps the operator quickly locate and correct problems.

The DLC processor models and predicts the motion profile, and continuously compares it with the actual response of the servo controller, thereby detecting irregularities in drive conditions, such as drive runaway or excess position lag conditions. Parameters allow the user to set the magnitude of certain variations, as required for the application, before an error is considered a fault condition.

Programming Structure

The basic program for standard motions is user programmed. The user prepares a program of up to 3000 lines/blocks, utilizing pre-defined commands. These commands, represented by three letter mnemonic codes, specify the function. The DLC, when used with the optional CTA keypad/display, guides the user for proper entry of the necessary data for each command/function utilized, such as, desired position, desired

velocity, etc. The DLC can be programmed to run up to three separate task simultaneously (multi-tasking). The DLC can be programmed with several sub-routines. The user can select a different sub-routine from the main program to run different applications. The user can customize the operation of the DLC control for any number of particular applications.

The user can download program blocks to the DLC from a host device (computer, PLC, etc.), while the control is in operation. Therefore, the effective program size can be much larger than 3000 lines, if needed.

Programmable Acceleration Rate

The acceleration rate, set by parameter, can be changed (reduced) by programming command. The rate can be changed to different levels for subsequent moves "on the fly" in automatic mode. This is useful for establishing proper rates for new materials or setting required rates for different materials without changing parameter settings.

Programmable I/O

The Standard DLC requires a DEA 4 card which includes a set of 7 auxiliary inputs and 11 auxiliary outputs which can be defined by the user for electrically controlling and acknowledging machine functions. Optional I/O cards can increases the auxiliary I/O to 37 inputs and 43 outputs. Additional outputs can be programmed as flags.

The Standard DLC Control is illustrated in Figure 1.5. The Extended version, with additional I/O connections, is illustrated in Figure 1.6.

Control/Machine System I/O Interconnection

The DLC has 8 input and 5 output connections which are pre-defined. They include connections to the machine and its control panel for mode selection, cycle start and stop, emergency stop, mode selection acknowledgment, etc. These connections are typically made to keep handshake between the control and machine. For example, on a slide, the control will not position if the ram is too close to the material, and/or the external operation will not start until the positioning is complete. The axis will not position if an external operation is pending.

Homing

Homing allows absolute referencing for the axis. The user can initiate homing in the manual mode or automatic mode of operation. The DLC offers a great deal of flexibility in customizing the homing routine to compensate for backlash, forward-moving-only applications, homing to a switch, or a variety of other needs.

Registration

Registration control maintains each position as close as possible to a registration mark printed on the material. This ensures that printed patterns are kept in alignment with the finished product. Registration accuracy is limited to 1 millisecond input acknowledgment time.

RS-232/485 Serial Interface

A multi-format serial interface allows communication with a programmable logic controller, a Indramat IDS or SOT, a personal computer or other host device. All information normally entered with optional CTA display can be communicated over the RS-232/485 Serial Interface at rates of up to 19200 Baud.

CTA Remote Keypad/Display

The CTA is a remote keypad/display which is mounted separately from the DLC. The DLC Digital AC Servo System is panel-mounted inside a

cabinet, with the CTA separately mounted on the cabinet surface. When using CTA 01.3-B, with backlit display, the maximum distance from the DLC card is 30 feet. When using CTA 01.3-N, with non-backlit display, the maximum distance from the DLC card is 90 feet.

Optional IDS Module

An optional thumbwheel switch module (IDS) with two digit, seven segment display, illustrated in Figure 1.3, are available for the DLC. The IDS connects to the RS-232 connector X31 of the DLC. This unit is remotely mounted, up to a maximum of 60 feet away from the DLC. The operator selects the required feed length and a feed rate on different sets of thumb-wheel switches. The decimal place (resolution) for the feed length is set by parameter. The feed rate is selected as a percentage of the maximum feed rate set by parameter. All status and diagnostic message codes appear on the two digit LED alphanumeric display.

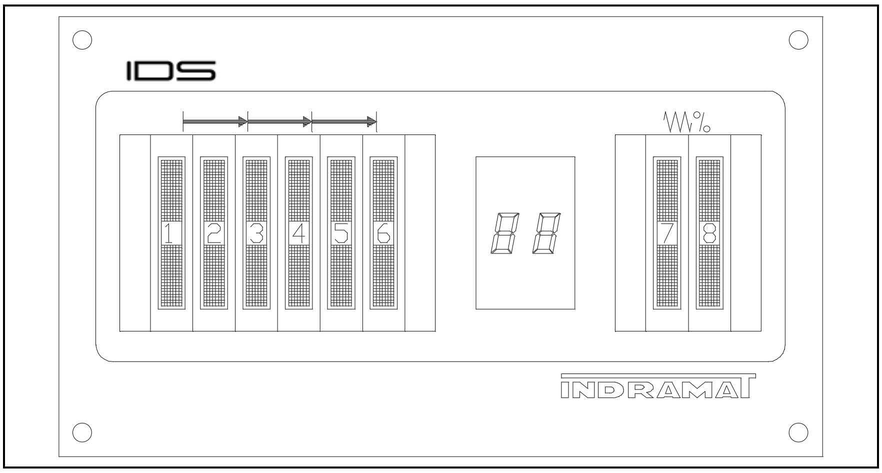


Figure 1-3: Optional IDS

Optional Station Operator Terminal (SOT) and Screen Manager

The Indramat SOT is a remote mounted, operator control device for the DLC (see Figure 1.4). It allows for the same input functions and displays the same information as the CTA keypad/display, but provides several additional features.

The SOT includes a backlit, liquid crystal display with 16 lines of 40 characters each. It can display much more information at a time than the standard display on the CTA keypad/display. The SOT keypad includes "click contact keys for entering or changing data. The SOT also has 8 outputs which can be defined by the machine builder with ScreenManager software.

ScreenManager software is a development tool for the SOT. This command line editor software package runs on any DOS-based computer. ScreenManager is used to create application specific screens for the SOT. With ScreenManager, the operator can view or edit machine information on the SOT screen, in easy to read, useful, information for each specific application.

The SOT connects to the serial communications port of the DLC and can be mounted up to 3000 feet away, when configured for RS-485 communication. An SOT User's Guide is provided with the option.

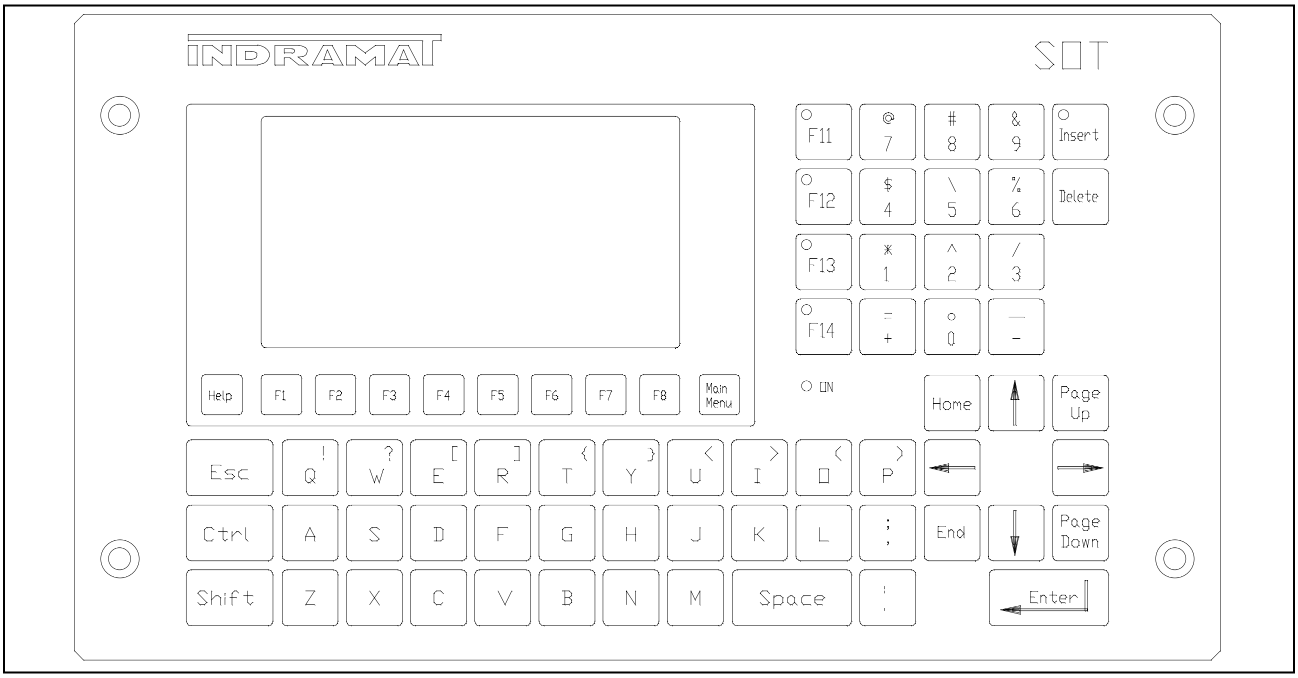


Figure 1-4: SOT - Station Operator Terminal

MotionManager™ (Option)

The MotionManager software development tool is an efficient method of creating and editing executable user programs for the DLC control. This user friendly software package runs on any DOS-based computer. It provides several benefits over programming the DLC from the CTA keypad/display. It also includes enhanced features for creating and editing programs that are not possible from the CTA keypad/display.

Optional CTA 10 User Interface

The CTA 10 is an optional user interface that can be used as a front end when a system contains multiple DLC cards. The serial port of the DLC must be configured as an RS485 station by using the CTA 4 keypad and an IKS745 cable. For more information, refer to the CTA 10 DL-1 User Manual.

Figure 1-5: CTA 10 User Interface

1.4 Standard Configuration Of DKS With DLC Control Card/DEA 4 Input/Output Card

The DLC Control Card plugs into the U1 slot and the DEA 4 Input/Output Card plugs into the U2 slot of the Indramat Digital AC Servo Controller. The Indramat Digital Controller being used mounts to the panel of a control cabinet (electrical enclosure). Installation procedures are described in Chapter 6.

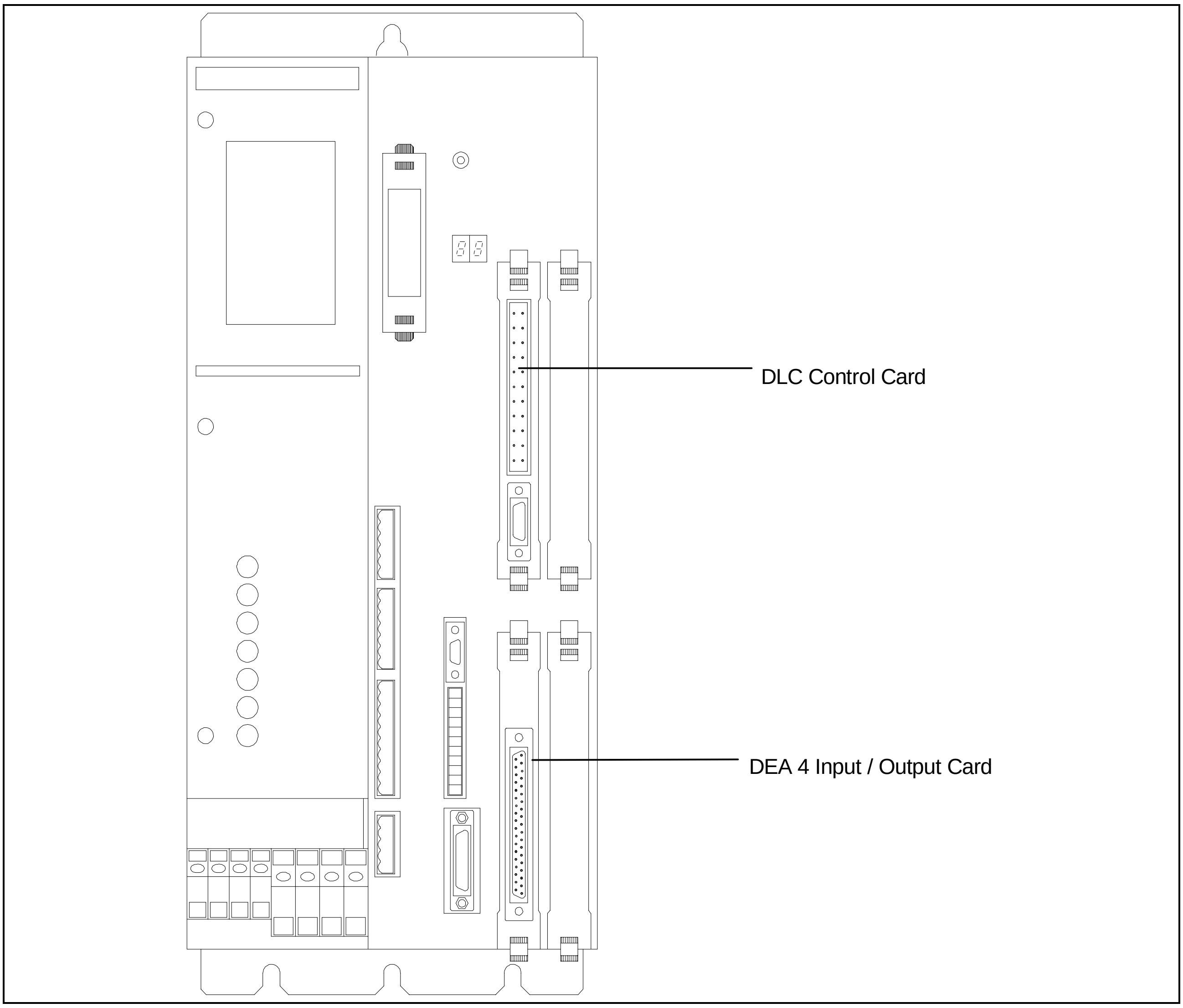


Figure 1-6: Standard Configuration Of DKS With DLC Control Card/DEA 4 I/O Card

The standard configuration for the DLC/DEA 4 with DKS (Digital Compact Drive) , illustrated in Figure 1.5. The DLC with DEA 4 Input/Output card has 15 inputs and 16 outputs. The first 8 inputs are system inputs and the next 7 inputs are auxiliary inputs. The first 5 outputs are system outputs and the next 11 outputs are auxiliary outputs. The auxiliary inputs and outputs can be defined by the user for controlling and acknowledging machine functions. The expanded configuration for the DLC/DEA 4/DEA 5/DEA 6 with DKS (Digital Compact Drive), illustrated in Figure 1.6, expands the auxiliary inputs to 37 and auxiliary outputs to 43. The functional description of the I/O signal connections is described in Chapter 3.

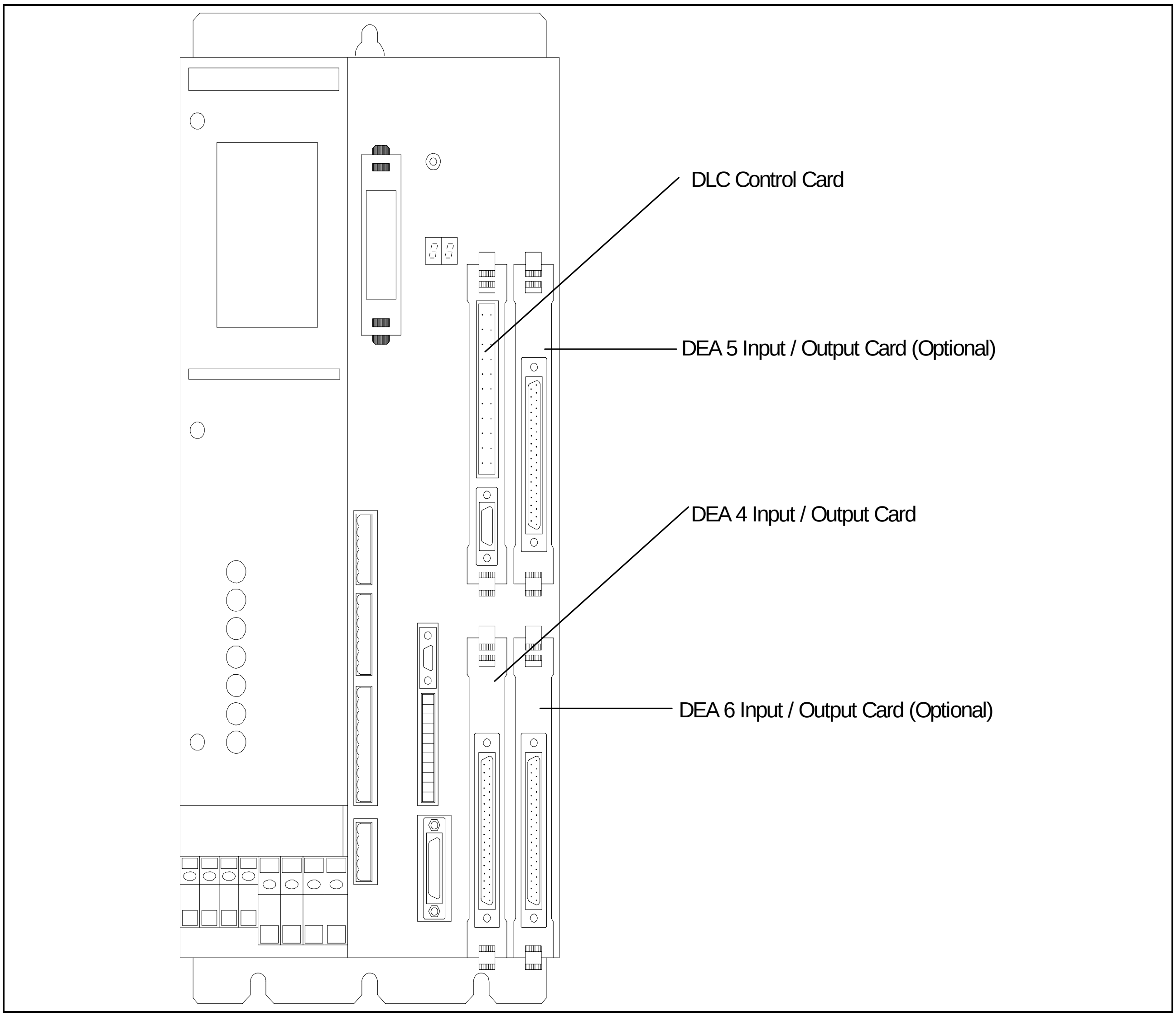


Figure 1-7: Expanded I/O Configuration Of DKS With DLC Control Card/DEA 4, 5, 6, I/O card

Standard Configuration Of DKS With DLC Control Card/DEA 4 Input/Output Card And Optional DEF 1 Incremental Encoder Interface Card

The DLC Control Card plugs into the U1 slot and the DEA 4 Input/Output Card plugs into the U2 slot of the Indramat Digital Drive. The optional DEF 1 incremental encoder interface card plugs into either the U3 or U4 slot. The DEF 1 card can be used as either a master encoder input or measuring wheel input.

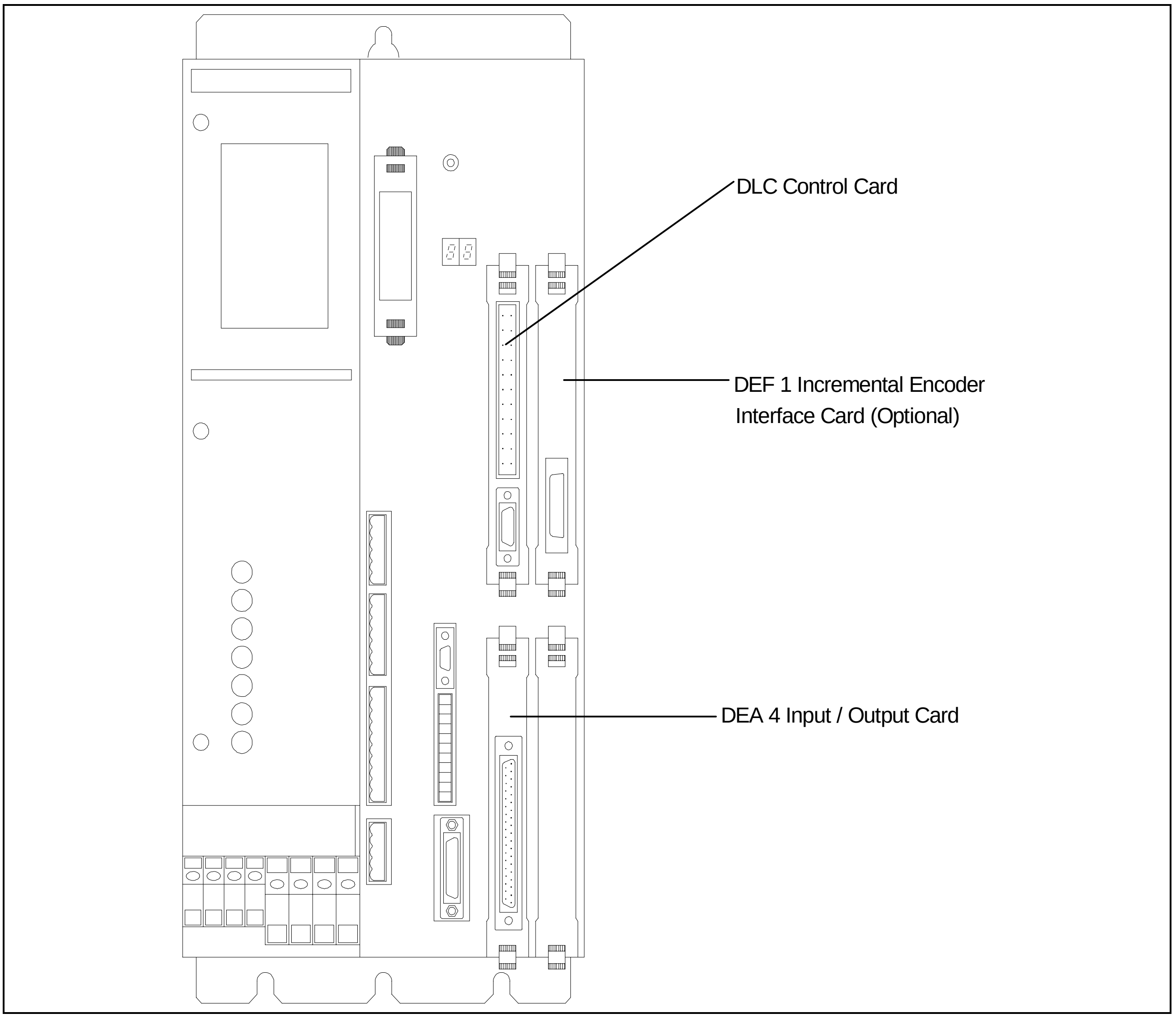


Figure 1-8: Standard Configuration of DKS with DLC Control Card/DEA 4 I/O Card and Optional DEF 1 Incremental Encoder Interface Card

1.5 Specifications

The following sections provide full specifications for the DLC Control and options.

Note: Performance specifications can vary, depending on the mechanical limitations of the equipment.

Physical Specifications--Operating Environment

Cooling	Convection
Allowable Ambient Temperature Range	41 to 113 °F (5 to 45 °C)
Storage and Transport Temperature Range	-22 to 185 °F (-30 to 85 °C)
Maximum Operating Altitude at Rated Values	3,280 ft. (1000 meters) (higher altitudes permitted with proper cooling)

Control Specifications

Position Feedback	High Resolution With Single-turn or Multi-turn Absolute, Resolver
Measuring Wheel Feedback	Incremental Encoder requires DEF incremental encoder interface card. Absolute encoder (GDM only) requires a DFF interface card.
Feed Length Resolution	0.001 inches (0.01 mm)
Feed Rate	Normal - 0.1 - 99.9% of Maximum Velocity DG: Feed Rate values can be in µpm (Operator Selectable) Jog - 0.1 - 99.9% of Maximum Velocity (Parameter Selectable)

Note: Maximum Feed Rate will vary, depending on the mechanical design of the equipment.

Jogging	Forward / Reverse (Manual Mode only)
Programmable Dwell Time	0.01 - 99.99 seconds in 0.01 steps
Programmable Counters	Limited only by number of program lines
CTA Keypad/Display	LCD (Optional Backlit Version Available, Four (4) line, 16 Characters/Line, 20 membrane switch keys.

I/O Interface

System Inputs	8 (+24 Vdc @ 10 mA) (pre-defined function)
Auxiliary Inputs with DEA card	7 - Standard 37 - Expanded (user defined and programmable)
System Outputs with DEA card	5 (+24 Vdc @ up to 100 mA, Sourcing) (pre-defined function)
Auxiliary Outputs	11 - Standard 43 - Expanded (User defined and programmable)

CAUTION: Inputs will have a 10 mA current draw at 24 Vdc. Outputs are thermally protected by a current limiter circuit which eliminates requirement for added fuses. If the load on the output causes a current draw in excess of 100 mA, the output bank of 8 outputs shuts off. The entire bank of 8 outputs on the DEA card must them be reset by cycling system power off and back on.

Options

CTA Remote Keypad/Display An IKS 745 cable allows remote mounting of CTA (keypad/display) to the machine's control panel.

RS-232/485 Interface Options This standard interface allows remote operation and other data transfer between the DLC and a optional host device, such as the IDS, SOT, computer or programmable controller

IDS Module A remote thumbwheel switch module used for entering feed length and feed rate for operation; displays status and fault codes via a two-digit LED.

SOT Station Operator Terminal- Used with ScreenManager to create application specific screens for displaying diagnostics, entering feed length, feed rate, viewing input/output status, etc.

2 Controls and Indicators

This chapter contains a general description of the DLC control layout, plus the following information:

1. Description of DLC with optional CTA keypad and display.
2. Description of the functions of the keys on the CTA keypad.
3. Description of display screens; how to scroll through different screens and how to interpret and change data on the screens.

The CTA keypad and display module (Figure 2.1) attaches to connector X30 on the DLC via 04-0745 shielded ribbon cable. The system input/output connections (via DEA 4.1 I/O card) are described in Chapter 3. The connections are further described in Chapter 6 for installation.

2.1 CTA Keypad and Display

The CTA keypad / display panel consist of a keypad with pressure-sensitive membrane type keys and a liquid crystal display (LCD) which shows up to four lines of 16 alphanumeric characters each. The number of lines and characters showing depends on the selected display mode and the current operating status of the control.

The display informs the operator of the operating status of the DLC system and displays all diagnostic messages. It is also used when entering or editing program or parameters from the keypad.

The keypad contains all the keys required for data entry, cursor movement, clearing fault/error messages, entering program and parameter data, etc.

The following sections describe the keypad and display functions.

The keypad and display module must be remotely mounted. The standard length for the IKS745 ribbon cable is 8 feet. Different lengths must be specified when ordering this cable. The maximum cable length is 90 feet..

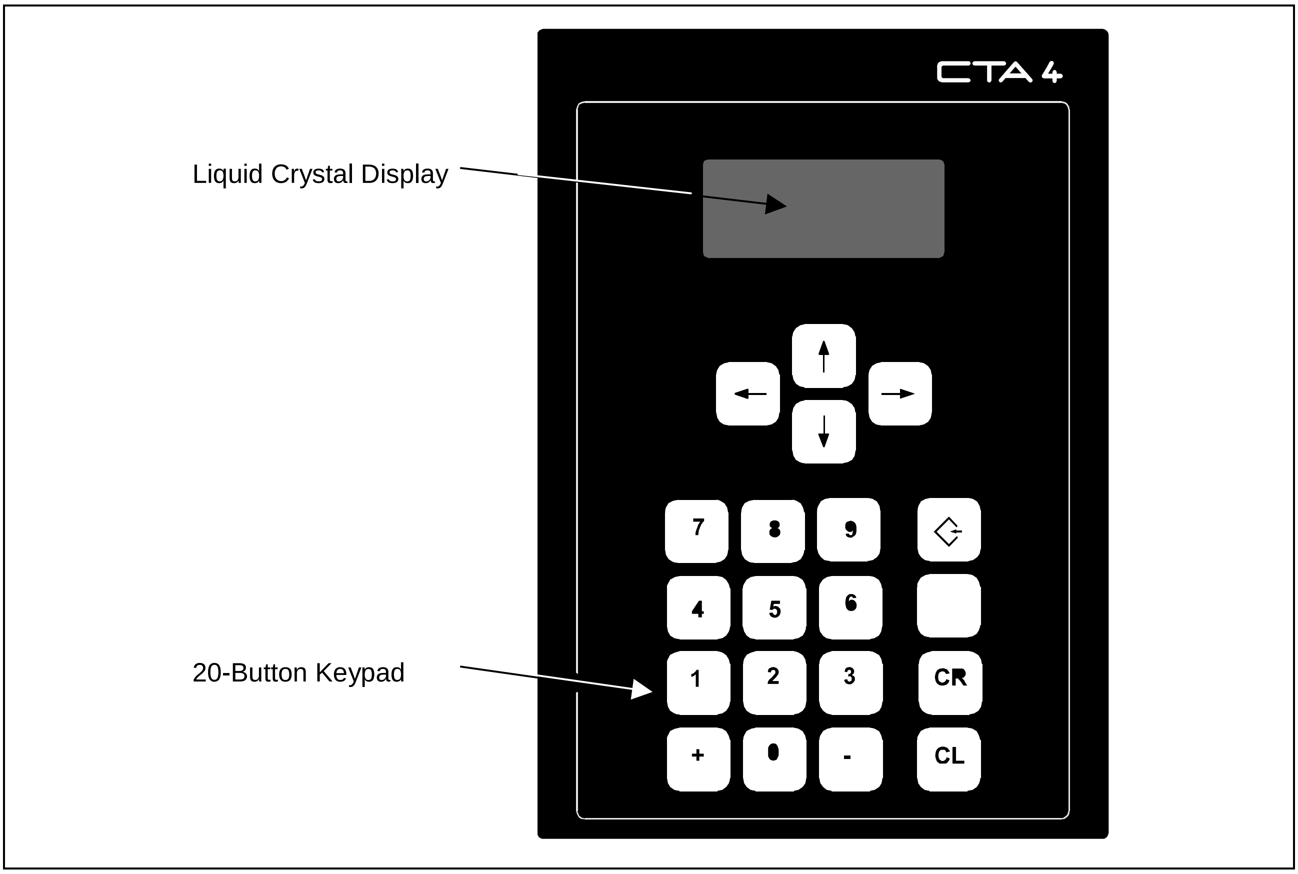


Figure 2-1: CTA Display/Keypad Module

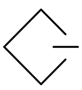
2.2 Data Entry Keys

This section describes the general function of each key on the CTA keypad. The use of these keys is further described throughout the manual for specific functions.

CL <u>Clear</u> -- Use to clear the displayed hard or soft fault message, if the fault can be cleared (cause of fault has been corrected). It also clears parameter entry errors. (See Store key for additional uses.)

CR <u>Carriage</u> <u>Return</u> -- When changing data values, press this key before pressing the Store key to cancel the change and leave the data as previously stored (clear entry).

In all display screens which show a flashing cursor (allow editing data fields), use this key to move the cursor to the first position of the data field; press again to move to a previous data field on the display. Continue pressing to move the cursor to home position of the display and allow scrolling to different display screens with the arrow keys.



<u>Store</u> -- Press to store (save) entered data to the DLC user memory when programming or editing program from the Edit display screen.

Pressing the CR key, changing to another block number or other display screen, without first pressing the Store key, cancels data changes and data returns to that previously stored.

+ & - <u>Plus</u> and <u>Minus</u> -- Use in programming (from Edit screen) to specify the feed direction. When on the Edit screen or Counter Display screen, use these keys to page through the block numbers (the cursor must be on the first line). Use in parameter mode to scroll through parameters numbers within each parameter set.

0 - 9 <u>Numerical</u> <u>Keys</u> -- Use for entering data values.

← → Left and Right Arrow -- Use to move the cursor to the left or right one position at a time. From certain display screens (ones without a cursor), the right and left arrow keys select additional display screens (see next section).

↓ ↑ Up and Down Arrows -- Use to scroll through display screens (see next section), or to change parameter sets (see Chapter 4) in parameter mode. Use to scroll through program commands when on the Edit screen. With the cursor positioned next to the command mnemonic (i.e. NOP_), press these keys to step through the program commands in alphabetical order.

Note: All displays illustrated in this manual use an underline character (_) to represent the cursor.

2.3 CTA Display Screens

The CTA uses its liquid crystal display for several screens. The operating mode and keyboard selections determine the resulting display.

When the DLC is in Parameter Mode, data for each parameter can be viewed, entered or edited. While in Automatic or Manual Mode, other display screens show the control software version, operation status messages, faults, status of each input and output, counters, etc. The Edit screen allows programming or editing the program data.

The following section describes procedures for scrolling through each of these display screens. Each following section describes the function of each screen, procedures to edit the screens data, etc.

Scrolling Through Display Screens

Refer to the "Display Map" in Figure 2.2 for a full illustration of the display access procedure. For convenience, the same illustration is included in Appendix C. This section describes the basic procedures for reading this "map" and scrolling through the different displays. Each display screen is fully described in the following sections.

To allow easier description, each row of the map is labeled A, B, C, etc. In general, use the up or down arrow keys to scroll through the different display screens. Use the left or right arrow keys to scroll through the displays on each row. All rows allow wrapping from the last screen on the row back to the first screen, and vice versa, by continuing to press the right (or left) arrow key.

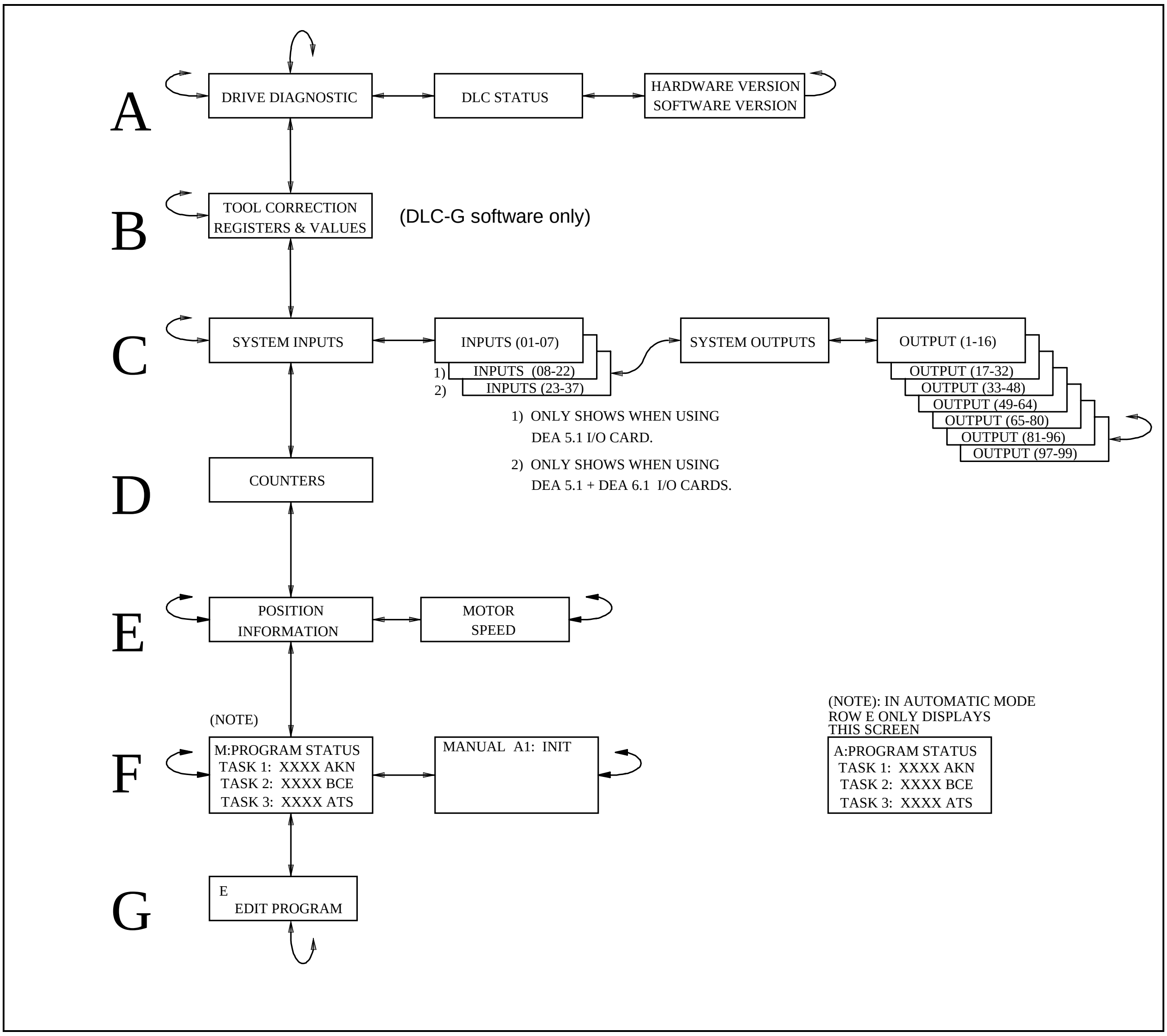


Figure 2-2: Map of DLC Control Panel Display Screens

Row A (Refer to the "Display Map" in Figure 2.2)

When power is applied and no errors occur, the DLC will display the Drive Diagnostic Screen in Automatic or Manual Mode, this display shows the drive status code and diagnostic. By pressing the right arrow key, the DLC control status is displayed. By pressing the right arrow key again, the DLC hardware and software screen will be displayed. This screen will show the current version of DLC hardware along with the current DG software version installed on the DLC control card. Use the right / left arrow keys to scroll around to different screens.

Drive Diagnostic 102 Drive Enabled	⇔	System is Ready	⇔	DLC 1.1 DG 01.1-XX.XX
Drive Diagnostic Screen		DLC Status Screen		Hardware/Software

If a servo fault is present at power ON, then the servo status message will appear first.

```
Drive Diagnostic
      26
Undervoltage
```

Example Drive Diagnostic Screen

If no servo fault is present at power ON, and a DLC fault is present, then the DLC status message will appear.

```
 EMERGENCY STOP
 EMERGENCY STOP
```

DLC Status Screen

Row B (Refer to the "Display Map" in Figure 2.2)

By pressing the down arrow key once from either the Hardware/Software Screen or Fault Screen, the Tool Correction Screen will be displayed (Row B). It shows the Tool Correction registers and their respective values.

Pressing the right or left arrow key causes the cursor to move through the data field and register number field. Tool Correction data is entered at this screen.

```
Tool Correction

Dnn ±00010.00
```

Row C (Refer to the "Display Map" in Figure 2.2)

By pressing the down arrow key once from the Tool Correction Screen, the System I/O Status Screen will be displayed (Row C). They show the status of each system and auxiliary input and output.

Pressing the right arrow key from the System Inputs screen causes the status of the Auxiliary Inputs to be displayed. Continue pressing the right arrow key to see the status of the System Outputs and Auxiliary Outputs. Use the left and right arrow keys to scroll through screens on the same line.

```
System Inputs     ⇔  Inputs     01-07  ⇔  Inputs     23-37
.11.1...             .....1.              ...............

System Outputs    ⇔  Outputs    01-16  ⇔  Outputs    97-99
1.1..                ..1.1......1....     .1.
```

System I/O Status Screens

Row D (Refer to the "Display Map" in Figure 2.2)

Pressing the down arrow key from any screen in the Row C, changes to the Counter Screen on Row D. It show the status of the counter now executing in the selected block number.

```
A:_0104 Counter
   000003  000005
```

Counter Screen

Row E (Refer to the "Display Map" in Figure 2.2)

Pressing the down arrow key from the display in Row D, changes to the Axis Information Screen in Row E. The Axis Information Screen consists of the axis position, axis following error, and the remaining distance to be positioned.

Pressing the right arrow key causes the Speed of Axis 1 to be displayed. This screen will display the commanded and actual speed of the axis in revolutions per minute (RPM).

```
Axis 1: Iner.
Pos.: +00001.000
FE. : +00000.010
Rem.: +00000.000
```

⇔

```
Speed Axis 1:

Com: 0500   RPM
Act: 0500   RPM
```

Axis 1 Information — Motor Speed of Axis 1

Row F (Refer to the "Display Map" in Figure 2.2)

Press the down arrow key from any display in Row E to see the Current Program Status. If the DLC is in Manual Mode, a "M" appears on the screen, along with the status of Task 1, 2 and 3. Use the left / right arrow keys to toggle between this screen and the Axis 1 Enabled screen.

```
M:  Progr.Status
Task 1:0091 BCE
Task 2:0366 VCC
 Task 3:0901 BCE
```

⇔

```
Manual  A1 Init.
```

Manual Mode Task Screen — Axis Enabled Screen

When the DLC is in Automatic Mode, only the task display is available on this row. An "A" appears on the screen (Automatic Mode), along with the status of Task 1, 2 and 3.

```
A:  Progr.Status
 Task 1:0091 BCE
 Task 2:0366 VCC
 Task 3:0901 BCE
```

Automatic Mode Task Screen

Row G (Refer to the "Display Map" in Figure 2.2)

From any screen in Row F, press the down arrow key to display the Program Edit screen in Row G. Notice the "E" in the upper left corner of the screen easily identifies it as the Edit display screen.

```
E _0096  BCE
0200  02  1
```

Edit Screen

Note: From Row G, press the down arrow key to wrap to a display screen in Row A.
From Row A, press the up arrow key to display the Edit screen in Row G.

Parameter Mode Display Screens

When the DLC is in Parameter Mode, data for each parameter can be viewed, entered or edited.

The top line of the display indicates the title of the parameter number. The following display shows Parameter A100 - Maximum Velocity for Axis 1.

```
Max Velocity

A100
00050.000
```

A100 - indicates parameter number

00050.000 - indicates data for parameter

When Parameter Mode is selected, the parameter display appears with a flashing cursor on the third digit of the parameter number. Use the right / left arrow keys to move the cursor in the field. Type over existing data to change and press the Store key to save the change. Appendix D provides blank parameter entry forms which show the required entry positions. Always maintain an accurate listing of your parameter entries for reference when troubleshooting or changing parameters for a different application.

There are two methods to select the different parameters numbers. The first method is to press either the left or right arrow key until the cursor is positioned on the parameter number and type in the desired parameter number. The second method involves the CR key on the CTA keypad. If the cursor is within the data field, pressing CR once will cause it to move to the beginning of that field. Press it again to move the cursor to the parameter number. Type in the desired parameter number. To change to different parameter sets, press the “Blank” key located just below the “Store” key. This will allow you to get to A1xx for Axis 1, B0xx for DLC system parameters, C0xx for DKS Drive Tuning parameters, and the C100 screen to reset DKS Drive to Standard Drive parameters. Refer to Chapter 4 for specific parameter entry options and requirements.

Drive Diagnostic/DLC Status/DLC Hardware/Software Version Display Screens

When the proper interconnection of the Indramat Digital Drive and DLC/DEA cards is completed, by turning on to the system power, the message “Waiting for Retrigger” will appear for 2-3 seconds, then the drive diagnostic screen will be displayed.

```
Drive Diagnostic
     102
Drive Enabled
```

Drive Diagnostic Screen

The DLC Status Screen displays the current operating status of the DLC control card screen In the example below, Automatic Mode is selected (A:) and Cycle Start has been pressed to start automatic operation.

```
A: Start Active
```

DLC Status Screen

If a hard or soft fault occurs, control function stops and a diagnostic message appears on the display. If a fault is present at power ON, a

diagnostic status message appears first (instead of the Drive Diagnostic Display).

See Chapter 8 for diagnostic messages and troubleshooting procedures.

```
EMERGENCY STOP
EMERGENCY STOP
```

Example Fault Status Display

By pressing the right arrow key, the current DLC hardware/DA software version installed in the Indramat Digital Drive will be displayed in Automatic or Manual Modes.

```
DLC 1.1

   DG 01.1-XX.XX
```

Hardware/Software Screen

DLC-1.1 - indicates hardware type

DG 01.1-XX.XX - indicates software number (XX.XX = revision number)

Press the left or right arrow keys to scroll through the Servo Status, DLC Status, and DLC Hardware/Software Version screens.

Tool Correction Data Screen (DLC-G only)

The tool length correction consists of a tool correction memory with 20 selectibale Tool Correction registers, D00...D19. D00 is effective upon entering automatic operation and is active once the program is started.

Inputting of correction values is possible on-line through the CTA keyboard. Tool Correction register values can be entered in manual and automatic mode. External Tool Correction data input is also possible through parallel inputs using the SO1 command (see Chapter 5 for a complete description of the SO1 command)

```
Tool Correction

Dnn  ±00010.
```

It is possible to use the Tool Correction registers for a variable memory, however, only one correction value can be selected and calculated per programming command.

Each correction value memory can accept a seven-digit correction value. The eighth digit is the sign bit (+/-).

System/Auxiliary I/O Status Screens

System/Auxiliary I/O Status screens show the status of each system and auxiliary input and output. These displays are helpful to verify wiring during start-up or troubleshooting. They also provide a quick summary of system/auxiliary status during normal operation.

System Inputs	⇔	Inputs 01-07	⇔	Inputs 23-37
.11.1...		.....1.		....

System Outputs	⇔	Outputs 01-16	⇔	Outputs 97-99
1.1..		..1.1......1....		.1.

System I/O Status Screens

The status of each of the input or output signal lines on a screen are represented by a character in the bottom row of the display. A Low signal (0 volts) is represented by a decimal (.). A High signal (+24 volts) is indicated by a one (1). The system Inputs and Outputs have a fixed or pre-defined function for the DLC. The first fifteen lines (inputs) of the DEA 4, connector X17, are for system and auxiliary inputs. The first eight are system inputs. The next seven are user-programmable auxiliary inputs. Two additional cards, DEA 5 and DEA 6, can be installed to expand the DLC auxiliary inputs up to 37 Chapter 3 provides a functional description for each of these I/O connections. Table 2.1 (following this section) provides a legend of the Input signal function for each Input screen, and the corresponding connector for hardware I/O connections.

The next sixteen lines (outputs) of DEA 4 connector X17, are for system and auxiliary outputs. The first five are system outputs. The next eleven are user-programmable auxiliary outputs. Two additional cards, DEA 5 and DEA 6, can be installed to expand the DLC auxiliary outputs up to 43. With either the Standard or Expanded DLC, these output points above the auxiliary connections (hardware), can be programmed as software flags. Certain output flags are set in firmware and can be queried by the user program. Chapter 5 defines the restrictions and use of these auxiliary outputs as software flags. Table 2.2 (following this section) provides a legend of the Output signal function for each Output screen, and the corresponding connector for hardware I/O connections.

CTA Display/Position	Signal Description	DEA 4 Connector pin
1	Parameter Mode Select	X17 1
2	Automatic Mode Select	2
3	$\overline{\text{Emergency Stop}}$	3
4	Cycle Start	4
5	$\overline{\text{Cycle Stop}}$	5
6	Jog Forward	6
7	Jog Reverse	7
8	Clear	8
9	Auxiliary Input #1	9
10	Auxiliary Input #2	10
11	Auxiliary Input #3	11
12	Auxiliary Input #4	12
13	Auxiliary Input #5	13
14	Auxiliary Input #6	14
15	Auxiliary Input #7	15
DEA 5 Auxiliary Inputs 8 - 22 (Requires DEA 5 Expanded I/O card option)		
1	Auxiliary Input #8	X32- 1
	through	through
15	Auxiliary Input #22	15
DEA 6 Auxiliary Inputs 23 - 37 (Requires DEA 6 Expanded I/O card option)		
1	Auxiliary Input #23	X33- 1
through		through
15	Auxiliary Input #37	15

Table 2-1: Input Signal Display Legend

CTA Display/ Position	Signal Description	DEA 4 Connector - pin #
1	System Fault Indicator	X17 16
2	Manual Mode Indicator	17
3	Automatic Mode Indicator	18
4	Parameter Mode Indicator	19
5	Auto. Cycle Running	20
Auxiliary Output 1-11		
1	Auxiliary Output #1	X17 21
through		through
11	Auxiliary Output #11	31
Auxiliary Output 12-27 (*Requires DEA 5 Expanded I/O card, see Table 5.1 in Chapter 5 for restrictions and use as software flags*)		
1	user defined	X32 16
through		through
16	user defined	31
Auxiliary Output 28-43 (*Requires DEA 6 Expanded I/O card, see Table 5.1 in Chapter 5 for restrictions and use as software flags***)**		
1	user defined	X33 16
through		through
16	user defined	31
Auxiliary Output Flags 44-59 (*See Table 5-1 in Chapter 5 for use as software flag***)**		
1	user defined as software flag	44
through		
16	user defined as software flag	99
Auxiliary Output 60-76 (*See Table 5.1 in Chapter 5 for use as software flags***)**		
1	user defined as software flag	60
through		
16	user defined as software flag	76
Auxiliary Output 81-88, 89-94, 95-96 (*See Table 5.1 for specific use and warnings***)**		
81	user defined as software flag	
through		
88	user defined as software flag	
89 - 94	function set in firmware as follows:	
89	1 indicates Manual Mode	
90	1 indicates Automatic Mode	
91	not currently defined/used	
92	not currently defined/used	
93	Warning	
94	0 indicates a system Fault	
95 - 96	function set in user program to provide specific function, as follows:	
95	Monitoring Window, Axis 1 turned: OFF=1, ON=0	
96	not currently defined/used	
Auxiliary Output 97-99 Monitor (*See Table 5.1 for specific use and warnings***)**		
97	not currently defined/used	
98	1=Axis 1 motion is interrupted	
99	not currently defined/used	

Figure 2-2: Output Signal Display Legend

Counter Screen

The Counter screen shows the current status of the counter now executing in the selected block number. Any counter programmed in blocks 0000-2999 can be monitored using this display.

```
M _1234 Counter
  123456  456789
```

Counter Screen

M- indicates operating mode (M=Manual Mode selected, A=Automatic Mode selected)

1234 - indicates block number of counter

123456 - actual number of counts

456789 - preset number of counts

Use the left or right arrow to move the cursor and enter a program block number which contains a counter command. The second line displays the current values of the counter. To scroll to the next or previous block with a counter, use the **+** and – keys. To leave the counter display mode, use the up or down arrow keys.

A counter can be programmed (see Chapter 5, BAC or COU command) to execute “X” number of parts then stop, and turn ON a light, etc. Count is maintained during shut-down, so the actual manufacturing process can be over several days. Counters can also be used to keep track of production.

NOTE: The actual count will be maintained by battery backup, even when power is turned OFF.

Axis Information Screen

The Axis Information Screen displays the current position (Pos.), current following error (FE), and the current remaining distance (Rem for Axis 1). By pressing the right arrow key the Commanded and Actual Speeds will be displayed in RPM (Revolutions Per Minute) for Axis 1. The values on these screens keep changing as they display the actual current system information. Each display screen is further described below. See following section 2.3.6.2 for changes in displays when the Measuring Wheel option is enabled.

```
Axis 1 Incr.
Pos.: +00001.000
FE. : +00000.010
Rem.: +00000.000
```

Axis Information

Pos. - indicates axis current position

+00001.000 -indicates direction of travel (+/-) and actual position in Input Units

The position values displayed can be cleared by pressing the CR key (if the axis has not been homed).

FE - indicates the axis current following error or position deviation.

+00000.010 - difference between the commanded position and current actual axis position in Input Units.

This value is directly proportional to the velocity command output to the amplifier. The higher the speed, the higher the following error. This value is zero only when the drive is in position. An excess following error (limits set by parameter) due to binds in the system, etc. will cause a fault condition.

Rem - indicates axis remaining distance

+00000.000 - difference between the commanded feed length and current actual axis position in Input Units.

The value displayed here is the distance remaining before the axis reaches the commanded target position. When move is complete and axis is in position, this value will be zero.

Actual Speed/RPM Display Screen

```
Speed Axis 1:

Com.:   0500 RPM
Act.:   0500 RPM
```

Actual Speed Screen

Com. - Commanded speed in RPM

Act. - Actual speed in RPM

Optional Measuring Wheel Encoder Position Screen

When the Measuring Wheel option is selected, the System Information and Actual Speed screens are expanded to include the Measuring Wheel Encoder Position Screen. The Measuring Wheel Encoder Position screen displays the current measuring wheel encoder position, current axis position and the difference between them.

```
L MR  +00059.990
   1  +00059.690
Diff. -00000.300
```

Current Measuring Wheel Position Display

Line 1: MR - Measuring Wheel Encoder Position

Line 2: 1 - Axis Position

Line 3: Diff. - Difference Between The Measuring Wheel Encoder Position and Axis Position

Mode/Tasks Screens

If the DLC is in Automatic Mode, an "A" appears on the screen, along with the status of Task 1, 2 and 3. If the DLC is in Manual Mode, two screens are available. On the first screen, a "M" appears on the screen, along with the status of Task 1, 2 and 3. Use the left / right arrow keys to toggle between this screen and the Axis Enabled screen. Each display screen is further described below.

Task Screens - Manual Mode

```
M:Program Status
 Task 1:0091 BCE
 Task 2:1366 AEA
 Task 3:2901 BCE
```

Manual Mode Task Screen

This display shows the current block number and related command mnuemonic for each task. Task 3 runs when the DLC is in manual or automatic mode. The information keeps changing as each block of the program executes. Task 2 starts running at the block number set in parameter B006, only in automatic mode. Task 1 always starts at block 0000, only in automatic mode. In manual mode, the display shows information for tasks 1 and 2 for the program block that was executing or next to execute during automatic mode cycling (or starting block).

Axis Enabled Screen - Manual Mode

```
Manual  A1 Init.
```

Axis Enabled Screen

A1 - indicated the axis is "ready" to operate

Init. - indicates axis has been homed

Task Display - Automatic

```
A:Program Status
 Task 1:0091 BCE
 Task 2:1366 AEA
 Task 3:2901 BCE
```

Automatic Mode Task Display

This display shows the current block number and related command mnuemonic for each task. Task 3 runs when the DLC is in manual or automatic mode. Task 2 starts running at the block number set in parameter B006, only in automatic mode. Task 1 always starts at block 0000, only in automatic mode.

During automatic operation (after Cycle Start), the information keeps changing as each block of program executes. If Cycle Stop signal goes low (switch is pressed), the drive immediately stops and the display shows information for the program block that was executing or just about to execute.

Edit Screen

Use the Program Edit screen to enter the complete executable program. Also use this screen to edit or review an existing program.

```
E _0096 BCE
0200  02  1
```

Edit Screen

E - indicates edit screen

0096 - block number, 0000-2999 is user selectable for edit and review - automatically increments to next higher number during programming

BCE - command mneumonic

0200 02 1 - data for the block command

When first scrolling to this screen, a flashing cursor appears in the home position (first digit of the block number). As with other display screens which include a cursor, press the left or right arrow keys to move the cursor within the screen. Also, press the CR key to move the cursor to the home position. Press CR again to position the cursor to the first position of the previous field. Continue pressing the CR key until the cursor is in home position or over the left digit of the block number, before using the up/down arrow keys to scroll to different display screens.

To select different program blocks of information to display (for review or edit), press the right/left arrow keys to position the cursor over the first digit of the block number and type over the existing block number. Command and data for this selected block will then appear. To scroll through the block numbers, press the CR key to locate the cursor in the top line, then use the + or - key to scroll through the blocks. The block number will increase or decrease accordingly and display the respective data for each block.

To enter program, start at block "0000" for task 1. Note that task 2 and task 3 programs start at the block number assigned by the user in parameter B006. If block "0000" is not displayed, press the CR key until the cursor is positioned on the first digit of the block number. Type the number "0000" to display that block number.

To enter or change a program command for each block, first press the right arrow key until the cursor moves to the right of the three digit program command mnemonic (i.e. NOP_). Press the up or down arrow keys to increment or decrement through the commands alphabetically. When the desired command appears on the screen, press the right arrow key. The cursor moves to the beginning of the second line where the data fields appear specifically for the selected command. If this block was not previously programmed with this same command, asterisks (*) appear in the digit positions where data must be entered. After entering and verifying the program block, press the Store key to save the programmed block to memory. The next program block number automatically appears, waiting to be programmed. Continue this process until all lines of user program are done. Note that the program can be edited or added to at a later time.

When changing data from the Edit screen, pressing the CR key before pressing the Store key, will cancel the change and leave the data as previously stored (saved). The display remains at the same block number and the cursor moves to the first position of the previous field. Changing to another block number, display screen, etc. without first pressing the Store key also causes loss of a change to a block's data.

Chapter 5 provides the information on commands required for creating the user executable program.

3 Functional Description of DEA 4 I/O Connections

The DLC motion control system is designed to function harmoniously within the machine builder's equipment design. The DEA 4 Input/Output card which plugs into the U2 slot of Indramat's Digital Drives. The DEA 4 provides several input / output signals to handshake between the DLC control and the machine builder's equipment. This chapter describes the functional operation of the interfacing inputs and outputs of the DEA 4 with DLC control card.

The first sections of this chapter describe the various interface functions in terms of the inputs / outputs involved with each. This includes the pre-defined I/O connections of the DLC/DEA 4, as well as certain functions which the user can select through parameter or programming, using auxiliary I/O connections. Certain auxiliary output functions are set in firmware, as described in Chapter 5 for user programming.

This first section is followed by an individual description of each pre-defined I/O signal, including name, pin assignments, and functional description. The designer utilizes these signals as necessary to implement the DLC for the application, including the design of the system control panel.

3.1 Signal Definitions

The states of the input and output signals described in this manual are:

- High = +24 Vdc
- Low = 0 Vdc

A signal line is described as "active high" when its associated action is initiated by a high (+24 Vdc) signal level. It is described as "active low" when its function is initiated by a low signal (0 volts). Active low signals are shown in this manual (when specifically referred to as a signal) with a bar (line) over the signal name, as an example, $\overline{\text{Emergency Stop}}$. This signal must be high during normal operation. If it goes low, the DLC initiates the actions required for an emergency stop function.

3.2 Interface Descriptions

The DLC input and output signals provide handshaking to/from the machine builder's equipment. Certain I/O signals should be considered as functional groups of signals, working in concert.

These functions include:

- Operating Mode Selection
- Safety Interlocks
- Normal Operation Signals
- Axis Homing
- Manual Operations
- Fault/Diagnostic Circuitry
- Feed Monitoring / Program Interruption
- Special Functions

Figures 3.1, 3.2 and 3.3 illustrate the DEA 4 X17 Input/Output, DEA 5 X32 Input/Output and DEA 6 X33 Input/Output connectors and each pin designation.

The following sections describe each group of interface functions, first listing the I/O signals involved (and section where each signal is individually described), then describing the general function of the signals.

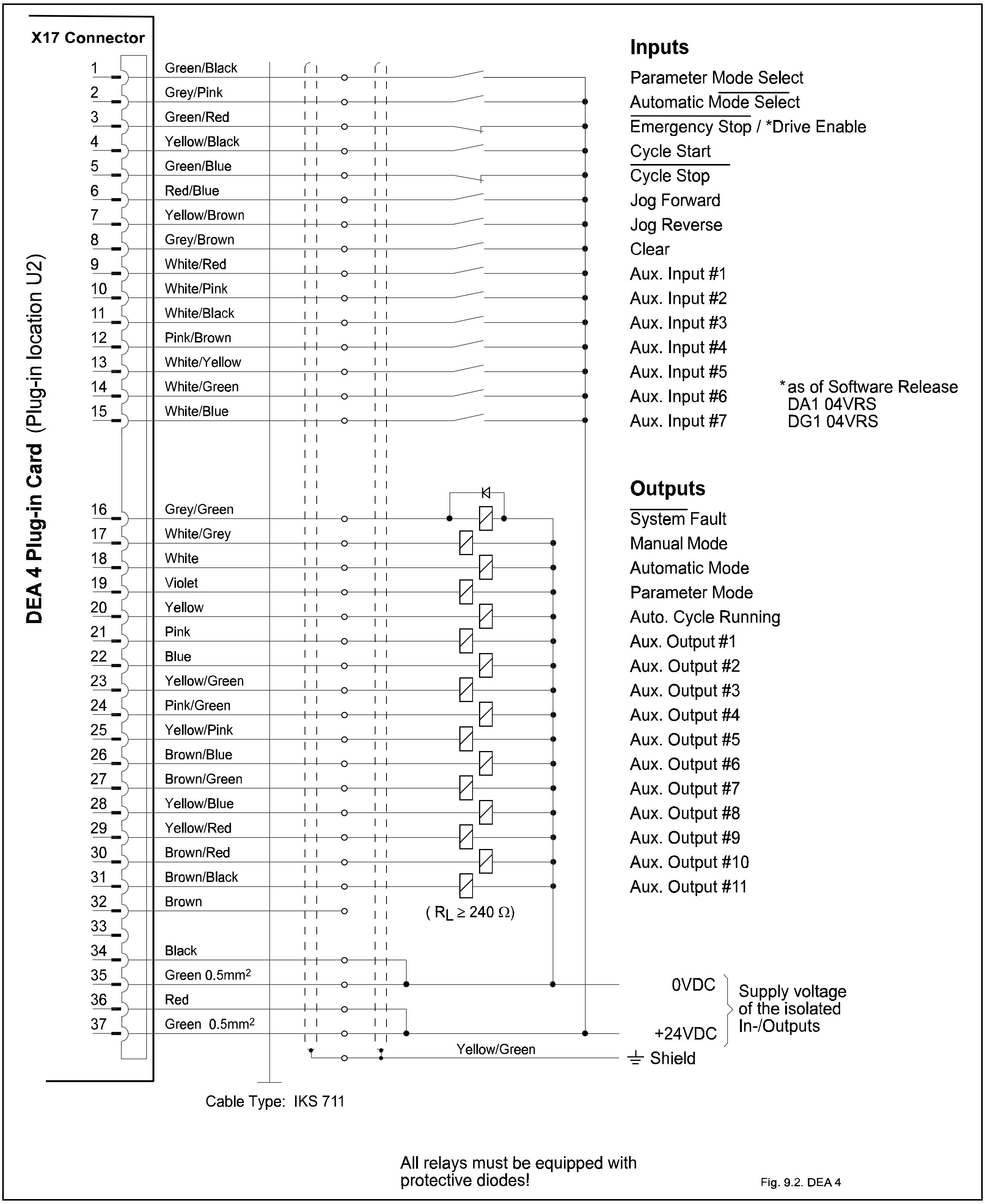


Figure 3-1: DEA 4 X17 Input/Output Connector and Pin Designations

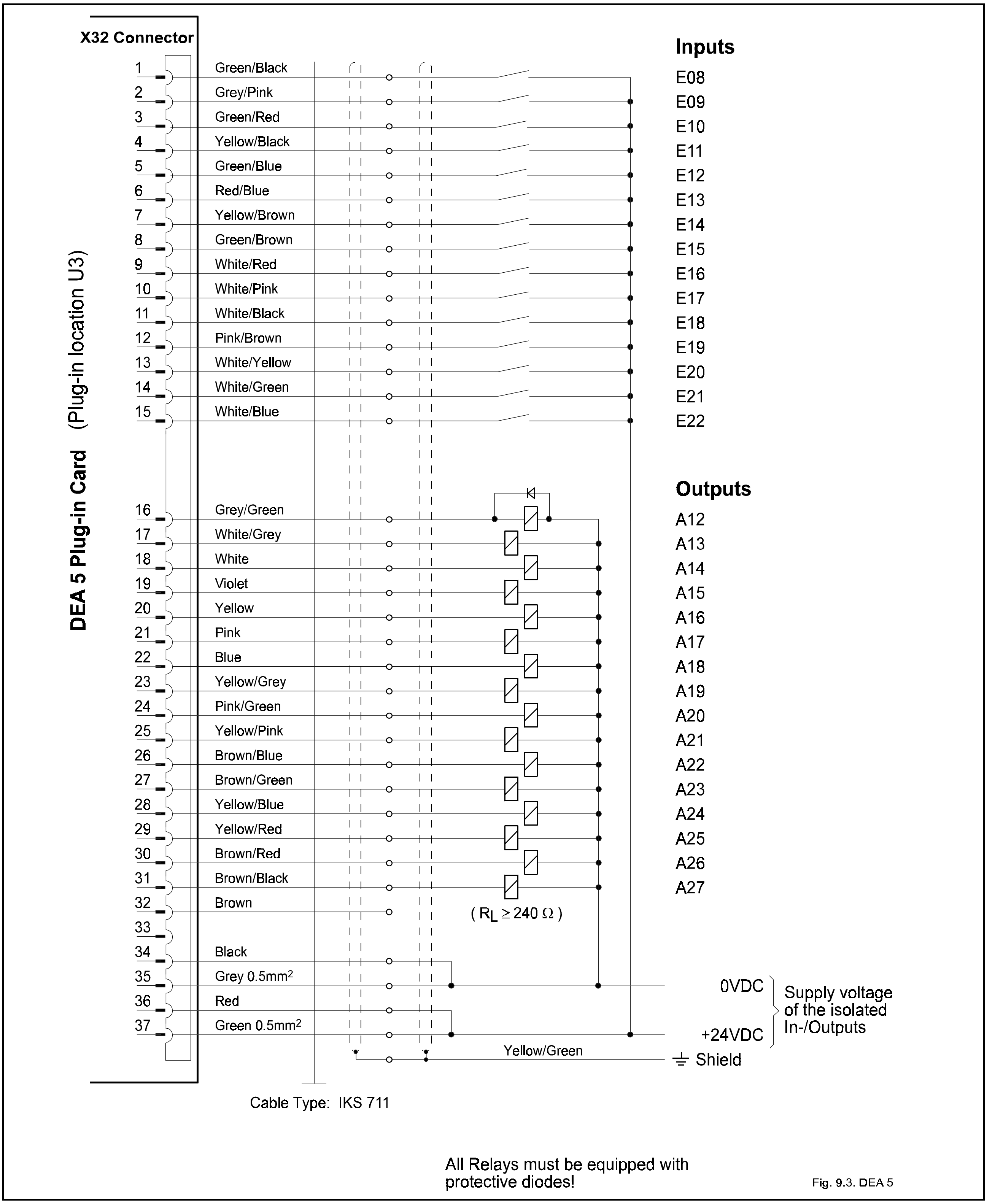


Figure 3-2: DEA 5 X32 Input/Output Connector and Pin Designations

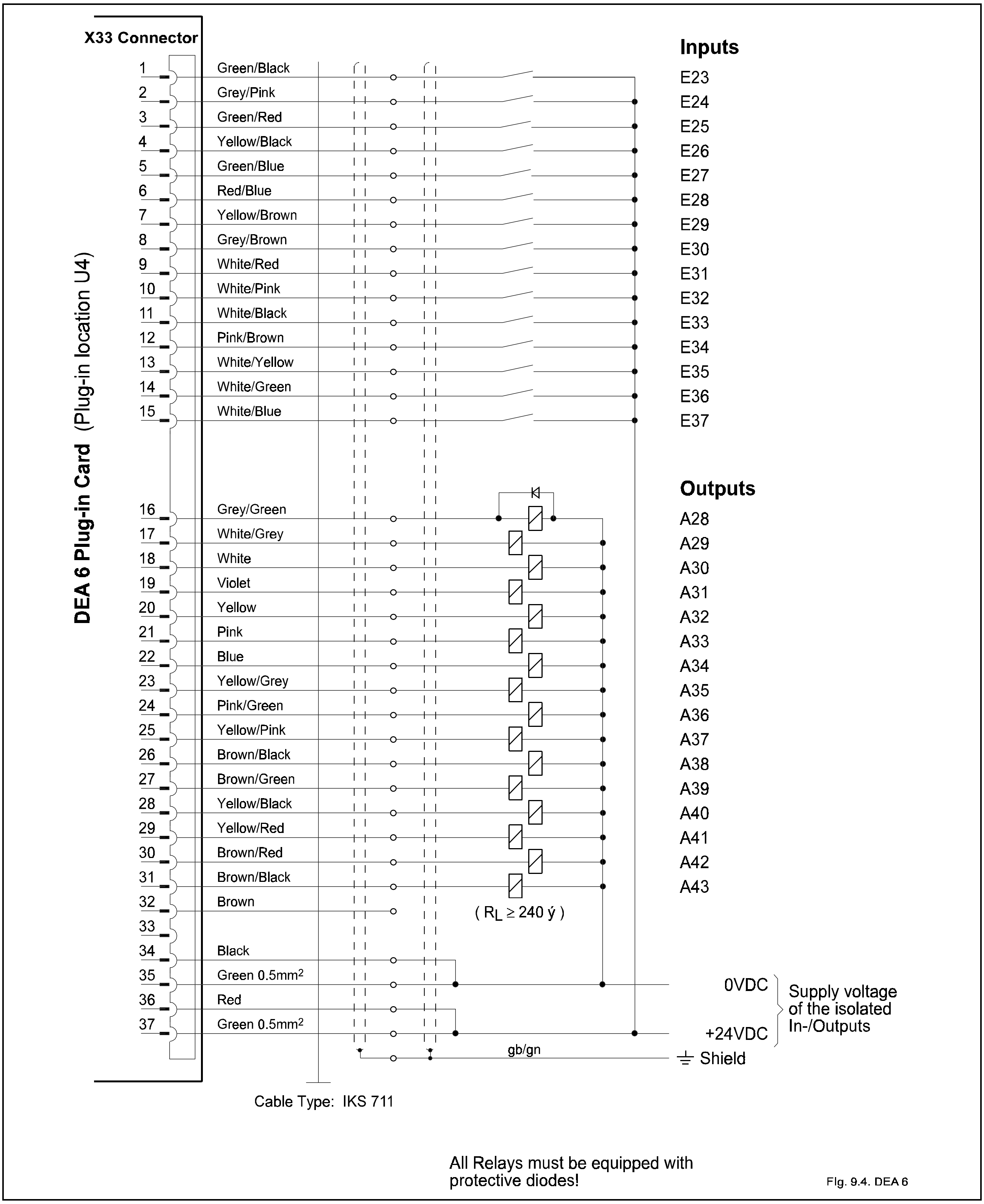


Figure 3-3: DEA 6 X33 Input/Output Connector and Pin Designations

Operating Mode Selection

Inputs

Parameter Mode Select (3.3)

Automatic Mode Select (3.3)

Outputs

Manual Mode Indicator (3.4)

Auto Mode Indicator (3.4)

Parameter Mode Indicator (3.4)

The DLC will always be in one of three operating modes:

- **Parameter Mode** - Allows entry/verification of the parameters required to adapt the control for the specific requirements of the application.
- **Automatic Mode** - Runs the user written executable program for automatic cycle operation (Task 1 and 2 run after a Cycle Start input).
- **Manual Mode** - Default mode when neither of the above are selected. In this mode, the jog inputs are used to position the axis forward or reverse, through the system.

Parameter and Automatic Modes are selected by bringing the appropriate signal line high (+24 Vdc). A fault is diagnosed and an "Invalid Mode Selection" error message is issued if Parameter Mode is selected while Automatic Mode is selected.

The suggested interface design is to wire the Parameter Mode signal to a key-switch, where a key is required to enter Parameter Mode and/or mount the switch inside the cabinet. This helps to prevent unauthorized parameter changes. Wire the Automatic Mode input signal to a two-position selector switch, where Manual Mode is selected when the switch is set to an open contact position.

The DLC has outputs to verify or acknowledge the currently selected mode. These are typically wired to indicator lights on the user's control panel or to a PLC.

Safety Interlocks

Inputs

- $\overline{\text{Emergency Stop}}$ (3.3)

$\overline{\text{Emergency Stop}}$ -- The $\overline{\text{Emergency Stop}}$ input must remain high for the DLC to operate. The system incorporates an Emergency Stop (E-Stop) chain. This is a circuit connected in series to both the DLC and the user's machine. Should any sensor in the E-stop chain open, all operations immediately stop. Note that the time taken to stop depends on the wiring method used, and inertia of the load connected to the servo motor on the machine. Note that task 3 of the user program continues to run during an $\overline{\text{Emergency Stop}}$ fault condition. When $\overline{\text{Emergency Stop}}$ is reset and a “CLEAR” signal is issued, task 3 starts at the beginning block and continues to run.

Elements connected in the E-stop chain commonly include the $\overline{\text{Emergency Stop}}$ switch on the user's control panel; E-stop switch(es) on the machine; switches on lubrication or coolant pumps; and various safety interlock switches on guards and doors.

Outputs

- $\overline{\text{System Fault}}$ Indicator (3.4)

There are several categories of faults, as described in Chapter 8. In general, once a fault is detected, an error message is displayed on the CTA display, the $\overline{\text{System Fault}}$ Indicator output turns OFF and all axes are immediately decelerated to a stop.

The fault recovery procedure is to first troubleshoot and remedy the problem. Then, press the Clear key on the DLC control panel (or external Clear input on user control panel, see section 3.2.6) to clear the DLC fault status and diagnostic message on the display.

Normal Operation Signals

Inputs

Cycle Start (3.3)

$\overline{\text{Cycle Stop}}$ (3.3)

The automatic execution of the user program begins when the Cycle Start line goes high (momentary) with the DLC in Automatic Mode. Once the automatic cycle begins, it is normally stopped by actuating the $\overline{\text{Cycle Stop}}$ input (signal goes low). The system will also stop if an error is detected or if the DLC is switched to Manual Mode.

Outputs

Axis Nearing Position (Pre Signal) (A107)

Axis In-Position (A106)

An auxiliary output (Position Pre-signal) can be assigned in parameter A107 to turn ON when the axis position is at a specified distance (in input units) from the target position of the commanded feed. Use this pre-signal when there is a need to anticipate the end of a feed, so another process can be initiated ahead of time. An example use would be to turn on a heater for bag sealing or plastic thermal forming operations. See Chapter 4 for further information on this parameter function.

An auxiliary output (Axis In-Position) can be assigned in parameter A106 to turn ON when the axis position is within the position threshold specified in this parameter. This position threshold does not affect the accuracy of the feed. It tells the program when to read the next program block (see PSI, PSA in Chapter 5). This output can be used to turn ON a light or buzzer on the user's control panel, or as a flag in the program to start a sub-routine program operation, etc. See Chapter 4 for further information on this parameter function.

Axis Homing

Inputs

Initiate Homing in Manual Mode (A112)

Home Switch (A112)

Outputs

Home Established Indicator (A112)

For absolute referencing for the axis , it is necessary to establish a correct measurement reference with the help of a homing routine. To initiate homing during manual operation, use auxiliary input signal (as defined in parameter A112); during automatic operation, by use of the "HOM" or G74 command.

Auxiliary inputs (Initiate Homing) can be assigned for axis 1 in parameter A112 to initiate the homing process in Manual Mode by a push-button switch or from a PLC output. Home position is set by a Home Switch. (Assign an auxiliary input for the home switch in parameter A112.)

Homing in Automatic Mode is accomplished through use of the HOM or G74 command. After reading in this command, the next command should be an ATS command to monitor the Home Established auxiliary output (as set in parameter A112) to prevent reading any Absolute Position commands (POA, PSA) until the home procedure is finished.

Homing Procedure--Refer to Figure 3.4

1) The axis moves towards the home switch, at the velocity set in parameter A110, unless the axis is already on the switch.

2) After the home switch closes, the axis decelerates to a stop on the marker pulse.

NOTE: For any amount of travel that exists in the minus direction of the home switch point, the home switch must remain closed. The home switch dog must be at least as long as the reverse travel distance from the home switch to the reading of the marker pulse.

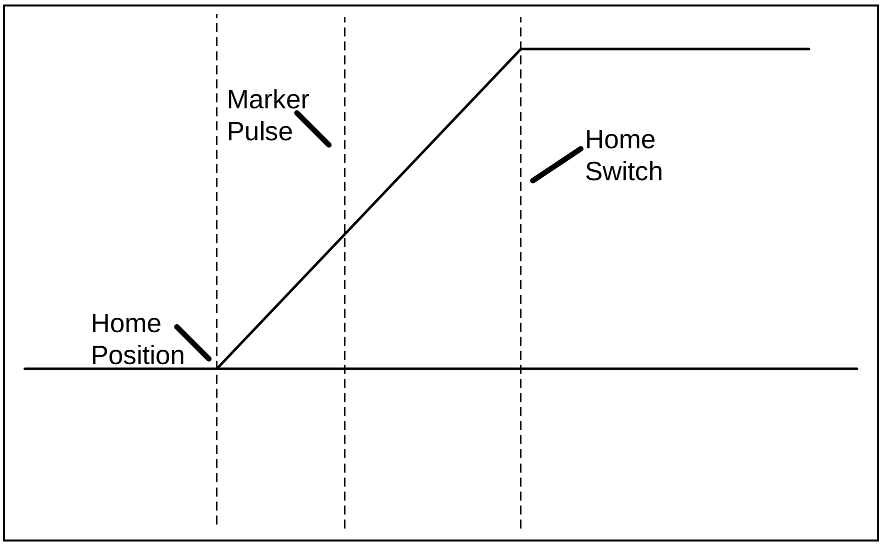


Figure 3-4: Axis Homing Routine

The zero pulse appears once in each motor revolution. The function of the Home Limit switch is to indicate the specific motor revolution in which the zero pulse is used.

In the homing procedure, regardless of the beginning position, the slide must stop on the Home Limit switch, closing it. The next zero pulse after the closing of the Home Limit switch indicates the reference position.

In many cases, some position other than home, such as the center line of the machine, is used as the reference position for machining. All programmed distances are then specified in reference to this point. This is established by entering the distance from home to the new reference location as the Reference Position (Homing Offset) in parameter A111. When the control is homed, this value will be loaded into the position counter, and all moves will be made in reference to this position.

Placement of the Homing Switch

As described above, the reference position of the AC servo drive is determined after it has moved onto the switch.

The reference position can be set in steps of one motor revolution each, by using the position the switch is in when it is activated. By monitoring the switching point of the homing switch and the marker pulse, a DLC diagnostic check eliminates the danger that these two are so close to each other that the switching tolerance limits will result in uncertainty about the motor revolution to be evaluated. If the switching point of the homing switch is closer than 1/16 of a motor revolution, with respect to the marker pulse, the control unit will not complete the homing process, and will change over to fault indication. The display will show: "Marker Pulse 1?." The homing switch should be relocated by 1/3 of the feed constant.

Activating the Homing Switch

Activating the homing switch must be done in such a way that the switch closes upon backing up, i.e. upon the movement away from the work piece. The dog cam for activating the homing switch must be long enough so that the activation is not canceled by a continuing reverse movement up to the minus travel limit of the axis. This is required to indicate to the control unit in which direction the slide must be moved in order to approach the reference position.

Since the interrelationship between the direction of rotation of a servo motor and the movement away from, and toward, the work piece depends on the particular design of the machine, the homing direction must be also set correctly in the parameters.

If there is no marker pulse within one revolution of the motor's feedback, homing sequence is halted, and a fault is displayed. The display will show: "No Marker Pulse 1."

Interruption of the Homing Routine

If, while in manual operation, there is a stop, an interruption, feed angle monitoring, or a switch of operating modes, the homing cycle is interrupted and must be restarted. After an interruption or stop occurs while in automatic operation, the homing cycle is restarted immediately by activating the Cycle Start input and initiating a HOM or G74 command in the program. After a fault or a change of operating modes during the homing cycle, the DLC must be re-initiated (switch to Auto Mode, press Cycle Start).

Over-Travel Switches

It is recommended that you install forward and reverse over-travel switches on applications that have limited axis travel, such as a slide or a ballscrew driven axis. The switches should be wired to the $\overline{\text{Cycle Stop}}$ input of the DLC. You must reserve distance beyond the limit switches to allow for the axis to decelerate to a stop.

Manual Mode Operations

Inputs

- Jog Forward (3.3)
- Jog Reverse (3.3)
- Initiate Manual Vector Program (B011)

The **Jog** inputs allow the axis to jog in the forward or reverse direction. Parameter A101 specifies the axis jog feed rate for Axis 1. It usually is set to approximately 10% of the maximum feed rate. It can never be more than the maximum feed rate.

In Manual Mode, the axis can be jogged forward or reverse. The Jog inputs are not functional in Parameter or Automatic Modes. A high on the Jog Forward input causes the axis to feed forward at the velocity set in parameter A101. The feed continues as long as the Jog Forward remains high. The Jog Reverse operates similarly, jogging the axis in the reverse direction.

In order to make use of the software travel limits when in Manual Mode, the axis must be homed first. It is recommended to prevent jogging by interlocking with the Home Established output (A112), until the axis home position is established.

An auxiliary input, **Initiate Manual Vector Program**, can be assigned in parameter B011 and used to initiate a user program to run in Manual Mode. This program must not contain any feed instructions and should not be located in the main program. Select by parameter to start this program by an external input (push-button switch) or automatically when mode is changed from Automatic to Manual. The program is aborted if switched out of Manual Mode. The manual vector is not accepted while jogging or homing in Manual Mode (see last section). Jogging or homing is not possible while the manual vector program is running. The manual vector program must end with an RTS command.

An example use of the Manual Vector program is to use the APE command to set the states of auxiliary outputs. This command is programmed to set a bank/group of ten outputs individually to ON, OFF or Don't Care states. This allows setting outputs for a "shut-down" mode, or as default before starting automatic operation. See Chapter 4 for additional information on parameters and Chapter 5 for programming commands.

Outputs

- Manual mode indicator

Refer to Section (3.2) Operating Mode Selection (Manual Mode Indicator (3.4))

Fault/Diagnostic Monitoring

Inputs

- Clear (External) (3.3)

The fault recovery procedure is to first troubleshoot and remedy the problem. Then, press the Clear key on the CTA control panel (or external Clear input on user control panel) to clear the DLC fault status and diagnostic message on the display.

Outputs

$\overline{\text{System Fault}}$ Indicator (3.4)

The DLC includes extensive diagnostic monitoring circuitry, detecting normal operating status, operator errors, errors in the control itself and machine faults.

There are several categories of faults, as described in Chapter 8. In general, once a fault is detected, an error message is displayed on the CTA display, the $\overline{\text{System Fault}}$ Indicator output turns off and the axis will decelerated to a stop. If the system is operating properly, the $\overline{\text{System Fault}}$ Indicator output will be on.

Feed Monitoring / Program Interruption

Inputs

- Interrupt Vector Program (B012)
- Feed Angle Monitoring (A120)
- $\overline{\text{Feed Interrupt}}$ (A120)

An auxiliary input, Interrupt Vector/Program, can be assigned in parameter B012 to interrupt the user program at any time and start an Interrupt Vector sub-routine program (Automatic Mode Task 1 only). When this input goes high, the current program sequence will be interrupted immediately or after the current subroutine program is finished (option set in parameter). The program sequence will then continue at the start program block number assigned in the parameter for the Interrupt Vector program. See Chapter 4 for further information on this parameter function.

An auxiliary input, Feed Angle Monitoring, can be assigned in parameter A120 to prevent the program from executing any position commands. The DLC will process all program blocks not containing any position commands. If there is no signal at the assigned input, the program continues to execute until it processes to a block containing a position command. The DLC will stop in this block until there is a signal at the input. If the input signal turns OFF during a feed, the feed will be stopped and an error message will be displayed.

An auxiliary input, $\overline{\text{Feed Interrupt}}$ ($\overline{\text{Feed Hold}}$) can be assigned in parameter A120 to prevent the program from executing any position commands. The DLC will process all program blocks not containing any feed lengths. If the input is low, the program continues to execute until it processes to a block containing a feed length. The DLC will stop in this block until the assigned input goes high. If the input goes low during positioning, the positioning will be stopped. Positioning will automatically resume as soon as the signal goes high.

An example of use for this signal is in conjunction with a material loop. Material is fed from an uncoiler into a looping pit, with an optical sensor in the pit tied to the $\overline{\text{Feed Interrupt}}$ line. If the material pulls too tight, the signal goes low and the cycle is interrupted. As soon as an adequate amount of material is fed into the looping pit, the signal returns high and the cycle immediately resumes. See Chapter 4 for further information on this parameter function.

WARNING: If the automatic cycle is interrupted by a low on the $\overline{\text{Feed Interrupt}}$ line, all motion is suspended. Once this signal is restored high, the cycle will immediately resume. The work area should not be entered if the motion has stopped as a result of this signal.

Special Functions

Inputs

- Detect Registration Mark
- Enable Measuring Wheel Operation and Dual Encoder Option

An auxiliary input, **Detect Registration Mark**, can be assigned in the REF command to cause the DLC to search for a registration mark on the material. To provide this input signal, use a sensor which detects a printed mark or stamp on each piece of material. A high signal must be present on this input when a reference mark is present under the sensor. The accuracy and duration of this signal is essential to the proper operation of position correction, especially if the mark needs to be detected at high speeds. The DLC can detect a registration input signal in 1 millisecond. Refer to the REF program command in Chapter 5 for additional information on using this feature.

For **External Encoder operation**, an optional encoder input card must be installed in the Indramat Digital Drive Product. Also, the external encoder option must be enable in parameters A123, B016 through B022. These parameters enable the external encoder function and establish the feed constant and the number impulses of the encoder. They also allow assigning an auxiliary input for selecting either the external encoder or motor high resolution feedback. During external encoder operation, the measurement of lengths is from the external encoder, rather than the motor's high resolution feedback.

The **Dual Encoder Option** in parameter A123 allows quicker response with external encoder operation. When selected, the position loop is closed using motor high resolution feedback and the external encoder provides adjustments to the length during feeding. When this option is not selected, the Position Loop is closed using the external encoder. Lack of stiffness in the mechanical components between the motor and the external encoder can require using a lower gain, which decreases the response time.

During manual operation, the position feedback always comes from the motor's high resolution feedback. During automatic operation, there is a choice between continuous external encoder operation and using an auxiliary input signal to select external encoder operation.

Error messages are:

a) Excess Pos. Lag

b) Drive Runaway

c) External Enc. Error

Outputs

- Achieved Programmed Velocity

The **Achieved Programmed Velocity** Output is an auxiliary output which can be assigned to turn on when the commanded velocity is equal to the actual velocity. This auxiliary output is programmed in the Special Function parameter A115.

3.3 DEA 4 Input Signal Descriptions

This section describes the system/auxiliary inputs to the DEA 4 I/O card for the DLC control system. The Input and Output signal lines of the DEA 4 I/O card are optically isolated from the internal bus structure to minimize electrical noise interference. If an I/O network option card is used for I/O, see machine builders I/O documentation for a complete listing of I/O used.

The connectors and pin numbers of these signals are described in the following sections and shown in Figures 3.1 DEA 4 (connector X17).

Parameter Mode Select

DEA 4 Connector	-	X17, pin 1
Function	-	(Input) Selects Parameter Mode
+24 Vdc	=	Parameter Mode selected.
0 Vdc	=	Parameter Mode is not selected - another mode is selected or the system has defaulted to the Manual Mode.

The Parameter Mode Select will override any other mode selection input. If properly wired, selecting Parameter Mode while Automatic Mode is already selected, does not cause an error. When Parameter Mode is deselected, the DLC returns to the previously selected mode - Automatic or Manual (default when no other modes are selected).

NOTE: Task 3 stops in Parameter Mode.

Automatic Mode Select

DEA 4 Connector	-	X17, pin 2
Function	-	(Input) Selects Automatic Mode
+24 Vdc	=	Automatic Mode selected.
0 Vdc	=	Automatic Mode is not selected - another mode is selected or the system has defaulted to the Manual Mode.

If this signal line is high at the same time the Parameter Mode is selected (input high), the DLC issues the "Invalid Mode Selection" diagnostic error message.

NOTE: Task 3 will be running in both Manual and Automatic Mode.

Emergency Stop

DEA 4 Connector	-	X17, pin 3
Function	-	(Input) Commands the servo system to stop immediately
+24 Vdc	=	Allows the DLC to operate properly.
0 Vdc	=	The servo system is commanded immediately to zero velocity. Drive reaches zero speed in the minimum time possible - given the inertia and maximum torque available. The DLC issues an "Emergency Stop" diagnostic message.

WARNING: This signal must be used to ensure safety.

WARNING: The E-Stop signal should go through the Zks input on the DKS drive (X9, pin 1 & 2) in series with this DLC input.

Conditions which warrant pressing the E-Stop include:

1. Any condition posing an immediate danger to personnel.
2. A jam in the machinery or any other condition that poses an immediate harm to the system equipment.

NOTE: The Task 3 program will continue to operate during an E-Stop condition. When the E-Stop condition no longer exists and the clear input is issued, Task 3 will jump to the first block in Task 3 and then run.

Cycle Start

DEA 4 Connector	-	X17, pin 4
Function	-	(Input) Starts automatic cycle
+24 Vdc	=	(momentary) Starts the execution of the programs in Task 1 and 2, when Automatic Mode is selected.
0 Vdc	=	Has no effect on the system operation. Once initiated, only a system failure (fault or error), the presence of Emergency Stop or Cycle Stop (or user defined Feed Interrupt) will halt the automatic cycle.

This input is typically wired to a normally open push-button switch on the user control panel.

NOTE: Task 3 will be running in both Manual and Automatic Mode.

Cycle Stop

DEA 4 Connector	-	X17, pin 5
Function	-	(Input) Used to stop the automatic cycle
+24 Vdc	=	(Continuous) Allows Automatic Mode operation.
0 Vdc	=	(Momentary) Stops the execution of the program in Task 1 and 2 in Automatic Mode.

This input is typically wired to a normally closed push-button switch on the user control panel.

When this input goes low during the execution of a program in Task 1 and 2, the program will stop in the block it is currently executing. When a cycle start is re-applied, the program will continue from where it stopped. If a position command is executing and a cycle stop is applied, the axis will immediately decelerate to a stop. The deceleration rate is the same as the acceleration rate.

NOTE: The axis cannot be jogged when this input is low.

Jog Forward

DEA 4 Connector	-	X17, pin 6, for Axis 1
Function	-	(Input) Jogs the axis in the forward direction
+24 Vdc	=	In Manual Mode, the axis feeds forward at the velocity set in Parameters A101. The axis will feed forward as long as the high signal is present (switch held closed).
0 Vdc	=	In Manual Mode, the axis stops feeding forward when a low signal is present (switch is released).

This input is typically wired to a normally open push-button switch on the user control panel.

Jog Forward is not functional when the DLC is in the Parameter or Automatic Mode.

Jog Reverse

DEA 4 Connector	-	X17, pin 7, for Axis 1.
Function	-	(Input) Jogs the axis in the reverse direction
+24 Vdc	=	In Manual Mode, the axis feeds in the reverse direction at the velocity set in Parameters A101. The axis will feed in reverse as long as the high signal is present (switch held closed).
0 Vdc	=	In Manual Mode, the axis stops feeding reverse when a low signal is present (switch is released).

This input is typically wired to a normally open push-button switch on the user control panel.

Jog Reverse is not functional when the DLC is in the Parameter or Automatic Mode.

Clear (External)

DEA 4 Connector	-	X17, pin 8
Function	-	(Input) Clears the Hard or Soft Fault status of the DLC
+24 Vdc	=	(momentary)- Clears the Soft Fault status of the DLC, or Clears the Hard Fault status and re-initializes the DLC.
0 Vdc	=	Has no effect.

Once a fault occurs, the DLC displays a diagnostic message (refer to Chapter 8). The operator must then physically correct the problem. Next, the operator can press the Clear key on the CTA keypad or the Clear pushbutton on the user's control panel, to clear the diagnostic message.

This signal is typically wired to a normally open push-button switch on the user control panel.

Auxiliary Inputs 1 Through 7

DEA 4 Connector	-	X17, pins 9 through 15
Function	-	(Input) Defined by the machine builder in the DLC parameters and program.
+24 Vdc	=	Auxiliary input is on (high).
0 Vdc	=	Auxiliary input is off (low).

The DEA 4 auxiliary inputs 1 through 7 are defined by the machine builder in the DLC parameters and/or program.

3.4 DEA 4 Output Signal Descriptions

This section describes the system/auxiliary outputs from the DEA 4 I/O card for the DLC control system. The Input and Output signal lines of the DEA 4 I/O card are optically isolated from the internal bus structure to minimize electrical noise interference.

The connectors and pin numbers of these signals are described in the following sections and shown in Figures 3.1 DEA 4 (connector X17).

NOTE: The current draw on any of these outputs should not exceed 75 milliamps or the entire bank of 8 outputs will shut off.

System Fault Indicator

DEA 4 Connector	-	X17, pin 16
Function	-	(Output) Indicates a fault has occurred
+24 Vdc	=	The system is functioning properly.
0 Vdc	=	The DLC has detected a fault.

This output is typically wired to an indicator light (ON when no fault is present) on the user control panel, or the signal is relayed to a buzzer when 0 Vdc occurs.

NOTE: Any Hard or Soft Faults will turn this output OFF. See Chapter 8 for a description of faults.

WARNING: This output is a semiconductor that should not be relied upon in the event of an emergency condition. If this signal is used, it should be in conjunction with the Emergency Stop chain.

Manual Mode Indicator

DEA 4 Connector	-	X17, pin 17
Function	-	(Output) Indicates that Manual Mode is selected.
+24 Vdc	=	No other modes are selected. The DLC defaults to Manual Mode.
0 Vdc	=	Another mode is selected or an error has occurred.

This output signal is typically wired to an indicator light on the user control panel or to a PLC.

Automatic Mode Indicator

DEA 4 Connector	-	X17, pin 18
Function	-	(Output) Indicates that Automatic Mode is selected
+24 Vdc	=	Automatic Mode is selected.
0 Vdc	=	Another mode is currently selected, or normal conditions that allow Automatic Mode operation are not satisfactory, or a fault is preventing entry into Automatic Mode.

This output signal is typically wired to an indicator light on the user control panel or to a PLC.

Parameter Mode Indicator

DEA 4 Connector	-	X17, pin 19
Function	-	(Output) Indicates that Parameter Mode is selected.
+24 Vdc	=	Parameter Mode is selected. This overrides any other mode selection.
0 Vdc	=	Parameter Mode is not selected.

This output signal is typically wired to an indicator light on the user control panel or to a PLC.

Automatic Cycle Running Indicator

DEA 4 Connector	-	X17, pin 20
Function	-	(Output) Indicates that Automatic Mode is selected and cycle start has been pressed.
+24 Vdc	=	Automatic Mode is selected and the cycle start has been pressed to start the user program in Task 1 and 2.
0 Vdc	=	Automatic Mode is not selected.

Automatic Mode is selected and cycle start has not been pressed. An error has occurred, preventing the operation of the automatic cycle.

This output signal is typically wired to an indicator light on the user control panel or to a PLC.

Auxiliary Outputs 1 Through 11

DEA 4 Connector	-	X17, pins 21 through 31
Function	-	(Output) Defined by the machine builder in the DLC parameters and program.
+24 Vdc	=	Auxiliary output is on (high).
0 Vdc	=	Auxiliary output is off (low).

The DEA 4 auxiliary outputs 1 through 11 are defined by the machine builder in the DLC parameters and/or program.

4 Parameters

This chapter describes the user-entered parameters required for the DLC to perform the motion control operation. The user adapts the DLC to his machine and the mechanical characteristics of the application by entering values for various parameters. These parameters permit a standard control system to conform to different but similar applications. It also assures that all application parameters are written with a uniform data format. The user must enter parameter values into the DLC memory prior to the operation and programming of the DLC control.

Note: All values for parameters must be known before an application program can be written. If a function is programmed or attempted which would exceed the bounds established by the parameters, the control will halt and a diagnostic error will be displayed.

4.1 Description of Parameter Sets

The DLC includes the following sets of parameters.

Parameter Set A (Axis Parameters):

This axis parameter set (A100 through A125) includes the axis operating values for Automatic and Manual Mode operation. These parameters allow the user to configure the DLC for the motor and drive package that the DLC is controlling. They also allow the user to set the various factors of the motion profile, as required for the material and application. These include the resolution of the feed, acceleration rate, feed rate in different modes, etc.

Parameter Set B (System Parameters):

This system parameter set establishes the operating arrangement of the DLC. It allows the user to configure the DLC for various options such as the language that shows on the display (English, German, Spanish French, Italian, or Portuguese); selecting control interface options, such as the IDS or SOT; selecting and configuring other optional features like synchronization, measuring wheel operation, multi-tasking, etc.

Parameter Set C (Drive Parameters):

This parameter set allows drive tuning through the DLC. Drive parameters can be tuned using the CTA keypad or RS232 serial interface. Default drive parameters can be set only by using the CTA keypad. Processing must not occur simultaneously over the CTA keypad and RS232 serial drive interface. Drive parameters are viewed and edited in the same manner as for A and B parameters. C parameters are administered in the drive. With each new power-up, the C parameters are sent from the drive to the DLC.

4.2 Parameter List

Table 4-1 lists all of the parameters for the DLC 01.1-A and the DLC 01.1-G with software versions DA01.1-XX.X or DG01.1-XX.X.

	Parameter Set A		**Parameter Set B (Continued)**
A100	Max Velocity	B007	Language
A101	Jog Velocity	B008	M-Function: I/O (DG only)
A102	Accel Rate	B009	M-Function: Timer (DG only)
A103	Position Gain	B010	Free
A104	Encoder Resolution	B011	Manual Vector
A105	Free	B012	Interrupt Vector
A106	Position Tol	B013	Override
A107	Pos Pre-signal	B014	Restart Vector
A108	Feed Constant	B015	Cycle Time
A109	Direction	B016	External Encoder 1
A110	Homing Setup	B017	External Encoder 1 Lines/Rev
A111	Homing Offset	B018	External Encoder 1Feed Constant
A112	Homing Ack	B019	External Encoder 1 Offset
A113	Min Travel	B020	External Encoder 2
A114	Max Travel	B021	External Encoder 2 Lines/Rev
A115	Special Funct.	B022	External Encoder 2Feed Constant
A116	Rotary Table	B023	External Encoder 2 Offset
A117	Knee Point		
A118	Free		
A119	Free		**Parameter Set C**
A120	Feed Angle Monitor	C000	Analog Output 1
A121	Max. Speed (RPM)	C001	Analog Output 2
A122	Monitor Window	C002	Overload Factor
A123	Follow Axis	C003	Position Data Scaling A-Output
A124	Free	C004	Velocity Data Scaling A-Output
A125	Jerk	C005	Velocity Loop Monitoring
A126	Position Window	C006	Position Velocity Window
		C007	Absolute Encoder Reference Position
	Parameter Set B	C008	Error Reaction
B000	Enable Axis 2	C009	Current Loop P-Gain
B001	System Inputs (DG only)	C010	Velocity Loop P-Gain
B002	User I/O (DG only)	C011	Velocity Loop I-Reaction Time
B003	Serial Interface	C012	Smoothing Time Constant
B004	Serial Interface	C013	External Brake Release
B005	Memory Display	C100	Set to Standard Parameter
B006	Start Task 2 & 3	C101	Set Absolute Position

Table 4-1: Parameter List

4.3 Entering the Parameters

The DLC must be in Parameter Mode to enter or edit parameters on the CTA keypad. The E-Stop signal must be present (+24 Vdc at DEA 4 connector X17, pin 3). The Parameter Mode is normally selected via a keyswitch on the user control panel, to prevent unauthorized changes to the parameter values.

Parameters are entered or changed by writing over any previous data and pressing the Store Key. These changes are made through the CTA keypad (or via a serial interfaced host device). A Lithium backup battery assures this memory is retained when the DLC is powered down or if the system experiences a power loss.

In Parameter Mode, the display shows the parameter name, number and data value of the parameter selected, as illustrated below.

```
Max Velocity

A100
00050.000
```

When selecting Parameter Mode, the display shows the first parameter of the **A** set of parameters (A100), unless a fault exists, as described later in this section. The parameter display appears with the cursor on the first digit of the parameter data field. Use the right / left arrow keys to move the cursor in the field. Enter or change a parameter's data by writing over any previous data stored. Press the Store key to save the data change. The display automatically changes to the next parameter number in the parameter set.

The following conditions will cause a parameter data change to revert back to the values last saved, if performed BEFORE pressing the Store key:

1. Pressing the CL key (clear entry),
2. Scrolling to another parameter,
3. Exiting Parameter Mode.

Always maintain an accurate listing of your parameter entries for reference when troubleshooting or changing parameters for a different application.

To select other parameters to display, first move the cursor onto the parameter number by pressing the CR key. If the cursor is within the data field, pressing CR once will cause it to move to the beginning of that field. Press it again to move the cursor to the beginning of the parameter number. The left and right arrow keys will also move the cursor within and between fields.

To change from one parameter set to another, move the cursor to the “A, B or C” parameter number and press the up or down arrow key. You can also press the Blank key below the Block Store key to scroll through the different parameter sets.

With the cursor on the parameter number, type over to enter the parameter number desired. The display changes as the new parameter number is entered. Pressing the + or - keys will also cause the parameter display to change to the next higher (+) or lower (-) parameter number within each parameter set.

Perform the following steps to change or edit the value of a parameter from the CTA keypad; see Chapter 7 for procedures to download from a PC or SOT via the serial interface:

1. Move the cursor (arrow keys) over the digit you want to change.
2. Enter the desired number on the CTA keypad (type over existing entry).

3. Verify data changes are correct in the display.
4. Press the Store key to save the new parameter values (displayed data) into memory. The display automatically changes to the next parameter.
5. Repeat this procedure to change each parameter required by your application. Maintain a current list of parameter entries. Appendix E provides work sheets which show the organization of each parameter and provide spaces to list each entry.

Note: Should the displayed message appear in a language other than the language desired by the user, enter the **B** Parameter Set and scroll to B007. Change the display to the desired language, as explained for parameter B007 in this chapter (field entry of 01xxxxxx will cause a display in English).

The DLC checks the parameter values each time it is powered ON, as well as each time parameters are read in. If there are parameters that are incorrect or missing, an appropriate error message will be displayed.

Upon exit from the Parameter Mode, if any parameters have been changed, internal buffers that are dependent on parameter values will be re-calculated. During that time the display will show the message "Please Wait!" If you have entered a value higher or lower than the limits of the parameter, the display will show the error message "Is INVALID." The invalid parameter description and number will display.

```
Is INVALID
Parameter A100

Max Velocity
```

By switching back to Parameter Mode, the invalid parameter will be displayed. Enter correct values for the parameter (within limits).

Displaying of Decimals

This manual describes a DLC software version that has programmable decimal positions. This version has two or three decimal places in the commands that involve positioning. Specific decimal places allowed for parameters and commands are listed in their specific description in Chapters 4 and 5. The number of decimal digits (can be up to 5) depends on the selection you set in Parameter B007. You must set this parameter before the other ones can be derived.

Note: If the decimal precision is not programmed in B007, "Is Invalid Language" diagnostic will appear when switched out of Parameter Mode. When switched back to Parameter Mode, the parameter with incorrect data is displayed.

Auxiliary Inputs/Outputs

Certain parameters require the selection of auxiliary outputs or auxiliary inputs (Acknowledgments) to be used. The auxiliary numbers to use must be selected by the machine builder and shown on their interconnect drawings, where applicable. Each auxiliary output selected should be unique; i.e. A106 and A112 should not use the same auxiliary output number.

WARNING: Auxiliary input and output numbers that have been dedicated by the machine builder for a specific purpose must not be changed. Personal injury or damage to the machine/drive train could result from such changes.

Unit of Measurement

The user specifies most parameter data in terms of input units (IU). A unit is defined by the user, and can be feet, inches, millimeters, degrees, radians, etc. Once the user has chosen the unit of measurement, then all position and feed rate data must be in accordance to that unit, where it is logically required.

For example, the input unit could be specified as inches. The maximum feed rate in parameter A100 will be programmed in inches per second. The acceleration parameter would then be in units of inches per second squared.

The F== feedrate command calculates the feed rate for you using the formula (Parameter A108 x 60). Therefore, all speeds programmed using the F== command are always in units-per-minute (upm).

Note that positioning commands for the application program are to two or three decimal places. If you require positioning to 0.001," set the decimal selection in parameter B007 to three places. In general, when using US System (inches) measurements, set B007 for three decimal places. When using Metric (millimeters), set B007 for two decimal places.

4.4 Linear or Rotary Operation

The DLC is capable of Linear or Rotary operation for each axis. Most applications will utilize the linear method of programming (i.e. Ball Screw, Belt, Slide). Some applications require the special features allowed with rotary programming (A116 - Rotary Table).

Linear operation (programming) allows each axis minimum and maximum software over-travel limits (effective after the axis has been homed). Motions can be programmed in any type of Input Units (i.e. inches, mm, degrees, etc.).

Rotary operation (programming) allows each axis the capability of 360 degrees of motion. The actual position screen (L P Screen), counts from 0 degrees to the value set in A108, resetting back to zero after each complete revolution (360 degrees). Each axis can be programmed to make multiple revolutions in either direction (direction is relative to the programming of A109 - Direction of Operation). The software over-travel limits are disabled when implementing rotary programming.

4.5 Parameter Descriptions

The following sections describe each parameter, as required for the DLC to command motion. Each parameter description includes an illustration of the CTA display and the formula required to obtain the parameter value, where applicable.

For easy reference, this section describes each parameter in numerical order, (A100 through A125, then B000 through B023, then C000 through C012), as shown in Table 4-1, and each new parameter begins on a new page.

Note: When entering the axis parameter data (A100-A125) and system parameter data (B000-B023), the Feed Constant parameter A108 needs to be determined before the velocity parameters can be entered correctly. The Feed Constant parameter (A108) sets the Input Units for positioning the axis.

Parameter A100 - Maximum Velocity

```
Max Velocity

A100
00050.000
```

Min.: 00000010 IU

Max.: 05000000 IU

This parameter defines the maximum velocity for the axis, entered in Input Units (defined in A108). All motion commands are a percentage of this maximum velocity. Velocity is specified in Input Units/second to two or three decimal places. Set the decimal resolution in B007 before entering this parameter. Motion commands programmed for 99.9% velocity results in a travel velocity equal to the maximum velocity (100.0%). All lesser numbers programmed, results in a travel velocity equal to the actual percent of maximum velocity entered.

The formula for calculating the max. velocity is:

$$\text{Max.Velocity} = \frac{\text{Max.RPM(A121)} \times \text{FeedConstant(A108)}}{60}$$

Notes:

Any gear ratio between the motor shaft (input shaft) and the output shaft should be taken into consideration in the calculation of the Feed Constant (A108). Refer to Parameter A116 - Rotary Axis Gear Ratio.

1. When using rotary table, the max. velocity is:

$$\frac{\text{Max.Velocity} = \text{Max.RPM}\left(\text{A121}\right) \times \text{FeedConstant}\left(\text{A108}\right)}{60} \div \text{GearRatio}\left(\text{A116}\right)$$

Example: Linear Maximum Velocity Calculation

Max. Motor Speed = 1500 RPM

Feed Constant = 2.441 IU's per rev.

$$\text{Max.Velocity} = \frac{\text{1500Rev.}}{\text{1Min.}} \times \frac{\text{1Min.}}{\text{60Sec.}} \times \frac{\text{2.441IU}}{\text{1Rev.}} \times \frac{\text{1500RPM}}{\text{60Sec.}} \times \text{2.441IU} = 61.025\,{}^{\text{IU}}\!/\!_{\text{Sec.}}$$

The entry in A100 would be <00061.025> for the system maximum,

or a percentage of this number for your preferred maximum velocity.

Note: Always round the result down; i.e. if the result was 33.333, the entry would be 33.332.

Parameter A101 - Jog Velocity

```
Jog Velocity

A101
00001.000
```

Min.: 00000010 IU

Max.: 05000000 IU

This parameter specifies the maximum velocity that the axis can be jogged in manual mode. It is specified in input units/second, to two or three decimal places. Set the decimal resolution in B007 before entering this parameter.

You specify this velocity in the same manner as in A100, but define the velocity to be used for jogging, rather than the maximum velocity.

Hint: Start at 10% of A100 entry, then increase or decrease as required for the application.

Parameter A102 - Acceleration Rate

```
Accel Rate

A102
0000386.0
```

The axis accelerates and decelerates at the rate specified by this parameter.

This parameter provides a protection for the machinery as it can limit the amount of torque produced during speed changes. The amplifier drive system must be capable of acceleration at the rate specified here. If not, an overshoot or an error message may occur during a speed change.

The parameter value is specified in input units $/second^2$ (units/second/second), in whole units (0 decimal places) or to one decimal place. Set the decimal resolution in B007 before entering this parameter.

For example:

To program the machine in the force of 1G,

assuming your Input Units are programmed in inches,

1G = 32.16 $feet/second^2$

Convert to inches, (input units, inches used in this example)

32.16 x 12 = 386.0 $inches/second^2$

For an Accel/Decel rate of 1G, enter parameter data <00003860>

Multiply or divide this number for a rate of 2G, 1/2 G, etc.

Use similar procedures to program in factors other than G force.

Note: Parameter A117, Knee Point, can be programmed to select a second acceleration / deceleration rate based on velocity.

Parameter A103 - Position Gain (KV Factor)

```
Position Gain

A103
0000 01.00
```

0000 - Not Used, set to 0000

01.00 - Position Gain

Minimum:	0.01
Maximum:	10.00

This is the axis position gain of the system.

The standard machining Kv Factor = 1. Entering a larger number (higher gain) will yield a tighter system (less following error). Entering too large a number can result in overshoot, oscillation, and/or high wear on the system parts (bearings, chains, gears, etc.). Set this parameter for optimum high performance (not necessarily maximum) for your system.

Input is specified to two decimal places.

Hint: A gain of 3 to 5 is typical when the motor is matched to the load inertia.

This entry sets the amount of velocity command given to the amplifier per a given position error (deviation). The following error (deviation) while feeding is inversely proportional to the KV selected.

$$\mathrm{KV} = \frac{\mathrm{Vel.}}{1000 \times \mathrm{FE}} \quad \text{or} \quad \mathrm{FE} = \frac{\mathrm{Vel.}}{1000 \times \mathrm{KV}}$$

Where:

- Vel (velocity) is in IU/minute
- FE (following error) is in IU (input units)
- KV=Velocity (in IU/minute) per thousand IU of Following Error

Parameter A104 - Encoder Resolution

```
Encoder

A104
00 000000
```

00 = Encoder resolution; sets maximum number of allowable motor turns.

00 = Default setting; ±65,536 motor turns

12 = 12 bit resolution; ±524,288 motor turns

13 = 13 bit resolution; ±262,144 motor turns

14 = 14 bit resolution; ±131,072 motor turns

15 = 15 bit resolution; ±65,536 motor turns

000000 = Not Used, set to 000000.

Note: This applies to single turn high resolution feedback only.

Parameter A105 - Free

```
Free

A105
00000000
```

Parameter A105, Free, is currently undefined.

Parameter A106 - Position Tolerance

```
Position Tol

A106
01 000.010
```

01 = Auxiliary output number to be turned on when position reached.

000.010 = Position tolerance to two or three decimal places. Set the decimal resolution in B007 before entering this parameter. i.e.. <XX000010 = 0.010 IU>

This parameter sets the axis positional band tolerance. It is defining a window in which the DLC will consider the axis in position. Adjusting this tolerance does not affect the accuracy of the move. It tells the control when to read the next command line (see PSI, PSM in chapter 5) and to turn ON the auxiliary output.

The first two digits specify the output number that will turn ON when the axis is in position. You can use this output to turn on a light, buzzer or as an internal flag. If no output is needed, a "00" can be programmed or an output that is not physically accessible (46 to 72) can be programmed.

Note: The auxiliary output number must be unique to all other parameter selected auxiliary outputs.

The remaining digits are the switching threshold (position tolerance), entered in input units. A typical setting is 5 thousandths of an inch. The second half of the entry would be <000005>.

The following sketch shows an **example** of switching threshold:

```
        Position Tolerance         |        |
        (in input units)           |        |
        at this point, program   ⇒|        |⇐
        reads in next block        |        |       |
                                   |      ⇒|       |⇐     Position Tolerance - The Aux. Output
                                   |        |       |       will turn back OFF momentarily if an
                                   |        |       |       overshoot exceeds this amount
   axis Direction  ⇒               |        |       |
------------------------------------------- 0 -------------------------------------------------- >
                                            |               overshoot (if any)  ⇒
                                            V
                                    Commanded Position
```

Note: A properly tuned system does not overshoot. See A102 and A103. If overshoot occurs, the acceleration rate and/or gain may be set too high.

Parameter A107 - Position Pre-Signal

```
Pos Presignal

A107
14 0.2 123.4
```

14 - auxiliary output number to be turned on (00=not activated)

0.2 - time in 0.1 seconds (0.1 - 9.9 in 0.1 increments)

00 = output as a constant signal which stays on until next feed or a command AEA (output ON/OFF).

123.4 - Pre-signal distance in Input Units (zero or one decimal place) to the target position. Set the decimal resolution in B007 before entering this parameter.

The Pre-Signal feature is used to turn ON an aux. output at a specified distance prior to the completion of a feed command. Typically, the Pre-Signal is used when anticipation of the end of a feed is needed, so other processes can be initiated ahead of time. The Pre-signal applies to each feed command: POI, PSI, POA, POM, PSA, PSM.

After the initiation of the motion, if the Current Distance to travel is less than the Pre-signal Distance programmed in this parameter, the Pre-Signal output is switched ON. The Pre-Signal output will be ON for either the programmed period of time or until the next move (dependent on the parameter settings).

```
                                                Pre-Signal Distance

                                              <-------------------------------->|

        Pre-Signal Off

¦ -------------------------------------------------------- ¦ -------------------------------- ¦
Begin Motion                                  Pre-Signal ON -        Target Position
                                              (For Programmed Time)
```

If the "Target Position" is equal to or less than the "Pre-Signal" distance, the Pre-Signal output is switched ON at the start of the feed motion for the specified amount of time.

```
¦ ---------------------------------------------------------------------------------------------- ¦
Begin Motion                                                            Target Position
Pre-Signal ON -
(For Programmed Time)
```

Parameter A108 - Feed Constant

```
Feed Constant

A108
001.00000
```

Minimum: 00001000

Maximum: 50000000

This parameter sets the feed constant, which is the ratio of slide movement per motor revolution.

The number entered equals the distance the axis will travel per one revolution of the encoder, to four or five decimal places. Set the decimal resolution in B007 before entering this parameter.

Note:	This parameter defines the input units used in the parameters.
Rotary Note:	Enter the amount of travel for one revolution of the table/device (one revolution of the gearbox output). The DLC will internally calculate the motor feed constant.

Parameter A109 - Direction of Operation

```
Direction

A109
0 1 000000
```

This parameter allows changing the axis direction (+/-) through software in the DLC.

0 = Not used, set to 0

1 = Direction

0 = Direction of operation remains unchanged

1 = Reverse motor direction

000000 = Not used, set to all zeros.

Parameter A110 - Homing Setup

```
Homing Setup

A110
0 1 0 1 05 0 0
```

This parameter specifies the search direction and speed for the zero reference point. See Section 3.2.5 for homing function description and Appendix A for additional application information.

0 - Not Used

1 - Search direction for home switch

0= Search Forward

1= Search Reverse

0 - Not Used

1 - Homing type

0= No homing (disable homing function)

1= Homing using a single-turn feedback (Incremental)

2= Multi-turn absolute feedback

05 - Search velocity in % of max velocity

01-99= % of A100 in 1% increments

(5% to 10% is normal for power up homing)

00 = When multi-turn absolute feedback selected, no search velocity is required.

0 - Not Used

0 - Linear/rotary operation select

0= Standard operation

1= Rotary operation, with shortest path

2= Rotary operation, follows programmed direction, forward/reverse

The last digit of this parameter allows using the DLC for rotary table applications. If a single-turn absolute feedback is used with this function, homing is required because of the absolute measuring system. The rotary table option also must be activated by entering a gear ratio in parameter (A116).

Parameter A111 - Homing Offset

```
Homing Offset

A111
0 0000.000
```

In many cases, some position other than the home position, such as the center-line of the slide, is used as the reference position. This parameter defines the distance (in input units), and the direction from the home switch that you want to use as the reference position. After the homing routine is completed, the offset distance programmed in this parameter will be inserted as the current axis position.

The axis will not physically move to this position. To physically position the axis to an offset position, after the homing routine is completed, use a position command (See Chapter 5).

This parameter is specified to two or three decimal places. Set the decimal resolution in B007 before entering this parameter.

For single-turn absolute feedback operation:

0 - Offset direction

0= Forward (+)

1= Reverse (-)

0000.000 - Offset distance in input units, to 2 or 3 decimal places

For multi-turn absolute feedback operation:

0 - No significance

0000.000 - Offset calculated from encoder reference point in input units, to 2 or 3 decimal places

For rotary applications, the zero offset may not exceed the value of input units / revolution of table. Otherwise, the error message "input error" will appear. For instance, if degrees have been selected as input units (360° / table revolution) and 10 has been programmed as the zero offset, the zero point is set at 10.

Refer to section 3.2.5 for additional information on the Homing function.

Parameter A112 - Homing Acknowledgements

```
Homing Ack

A112
09 12 00 13
```

This parameter is used to identify the various I/O assigned to the Homing Process. In Manual Mode, homing can take place by means of the assigned auxiliary input number, or in Automatic Mode by use of the command "HOM."

09 - Auxiliary input number used to initiate Homing in Manual Mode (push-button or PLC output)

00 = Manual Mode homing not used

01-37 = Selected input number to be used to initiate Homing

12 - Home Zero Reference Select (Marker Pulse or Home Switch)

00 = Homing to Marker Pulse for Single-Turn Absolute Feedback (Homing Speed for Homing to Marker Pulse is 3% of Max. Velocity)

01-37 = Selected input number to be used for home switch input

Note: The standard DLC with the DEA 4 I/O card has a maximum of 7 auxiliary inputs that are physically accessible to the user. With the optional DEA 5 and 6 I/O cards, the auxiliary inputs are expanded to 37.

00 - Unassigned

13 - Auxiliary output number to be used for "Home Established" signal

Note: The standard DLC with the DEA 4 I/O card has a maximum of 11 auxiliary outputs that are physically accessible to the user. With the optional DEA 5 and 6 I/O cards, the auxiliary outputs are expanded to 43.

For multi-turn absolute feedback operation, A112 should be set to 00000000.

Note: The auxiliary output for Homing complete/established can be turned OFF with an AEA command (output ON/OFF). The travel limits will remain in effect (still referenced to home).

Parameter A113 - Travel Limit, Minimum Value

```
Min Travel

A113
-01234.567
```

This parameter specifies the travel limit value in the negative direction, in reference to the Home Position. The limit is effective only after the axis has been Homed.

In the Manual Mode, the corresponding Jog key is disabled when this position has been reached. If, in the Automatic Mode, the commanded position is smaller (more negative) than this limit value, an error message will be displayed.

The travel limit value is measured from the reference point, Home Position, and is not added to, or subtracted from, the offset distance.

This parameter is specified in input units to two or three decimal places. Set the decimal resolution in B007 before entering this parameter.

For multi-turn absolute feedback operation:

A plus sign will appear for this parameter (instead of a minus sign) and the value here will be the minimum travel allowed. The travel limit value is measured from the zero point of the encoder and is not added to, or subtracted from, the offset distance.

Parameter A114 - Travel Limit, Maximum Value

```
Max Travel

A114
+01234.567
```

This parameter specifies the travel limit value in the positive direction, in reference to the Home Position. The limit is effective only after the axis has been Homed.

In the Manual Mode, the corresponding Jog key is disabled when this position has been reached. If, in the automatic mode, the commanded position is greater than this limit value, an error message will be displayed.

The travel limit value is measured from the reference point, Home Position, and is not added to, or subtracted from, the offset distance.

This parameter is specified in input units to two or three decimal places. Set the decimal resolution in B007 before entering this parameter.

For multi-turn absolute feedback operation:

The travel limit value is measured from the zero point of the encoder and is not added to, or subtracted from, the offset distance.

Parameter A115 - Special Functions (Disable Following Error, Velocity Achieved/Master Encoder Averaging/Velocity Feed Forward)

```
Special Function

A115
12 34 56 7 8
```

12 - Aux. Input to disable following error (01-37)

34 - Aux. Output to acknowledge following error is disabled

56 - Aux. Output to indicate when the programmed velocity has been achieved.

7 - Not used - Enter 0

8 - Averaging of Master Encoder input pulse train for Master/Slave Mode (See FOL command).

0 = No Averaging, No Velocity Feed Forward (1-9= Velocity Feed Forward Mode)

1 = Average over 1 read

2 = Average over 2 reads

3 = Average over 4 reads

4 = Average over 8 reads

5 = Average over 16 reads

6 = Average over 32 reads

7 = Average over 64 reads

8 = Average over 128 reads

9 = Average over 256 reads

Note: 1 Read = 1 millisecond. Feed forward automatically adjusts for following error between the axes, when averaging is selected. Any remaining following error can usually be minimized by slightly changing A121 RPM/10V (typical adjustment is less than 2%).

If no Special Functions are used, enter <00000000>.

Parameter A116 - Rotary Axis Gear Ratio

```
Rotary Table

A116
1234 5678
```

Use this parameter for Rotary applications.

This parameter sets the gear ratio of the input motor shaft to the output. The rotary table option is activated in the last two digits of parameter A110.

1234 - Input motor shaft turns (gear box input-encoder)

5678 - Output turns (gear box output-table)

Step down ratio = 1234 / 5678

The result per table revolution is:

Input units / Revolution of table = Feed constant x Step down ratio

Examples:

00010001 = a 1:1 ratio (1 turn input, 1 turn output)

00120001 = a 12:1 ratio (12 turns input, 1 turn output)

02300025 = a 9.2:1 ratio (230 turns input, 25 turns output)

Note: Gear down only; entry must be whole numbers. If the ratio is decimal, simply enter the number of teeth on the input and output gear shafts.

Parameter A117 - Second Acceleration Rate

```
Knee Point

A117
12 00386.0
```

12 - Velocity change point (in % of Velocity Max - A100) - Above this %, the second acceleration turns on - Knee point of acceleration change.

00 = Disables this parameter

01-99= % in increments of 1%

00386.0 - Second acceleration - Input Units/second² (units/second/second) to zero or one decimal place, above the knee point velocity. Set the decimal resolution in B007 before entering this parameter.

The first acceleration below the knee point velocity is programmed in parameter A102.

This function is effective in Automatic operation as well as in Manual operation (for instance, jogging). The only limitation for the user programming is that the command ACC (change of acceleration) is not effective as long as this parameter is switched ON (can use one or other).

CAUTION: Since the second acceleration may be greater than the first acceleration, the greater of the two values is to be considered the maximum acceleration.

Parameter A118 - Brake Release

(as of software version DA1-04VRS and DG1-03T03A)

```
Brake Release

A118
xx yy 0000
```

To be functional, this parameter must be enabled in Parameter C013.

xx- Input ‘Brake release’

00- External brake release function disabled.

yy- Output ‘Brake released’

00- No output programmed.

The output is functional only if the input ‘Brake release’ is also programmed.

Parameter A119 - Free

```
Free

A119
00000000
```

Parameter A119 Free, is currently undefined.

Parameter A120 - Feed Angle Monitoring, Feed Interrupt

```
Feed Angle Mon

A120
01 02 03 00
```

01 - Aux. input number for Feed Angle Monitoring with error display (01-37)

00 = Feed Angle Monitoring not used

02 - Aux. input number for Feed Interrupt (Feed Hold) (01-37)

00 = Feed Interrupt not used

03 - Aux. output message for Axis Stopped (01-99)

00 = Stop Axis not used

00 - Not Used

Note: There are no diagnostics for Feed Angle input missing before a feed or when a Feed Interrupt is executed.

1. Feed Angle Monitoring (Active High)

This parameter specifies if the feed or positioning is to be monitored or not. If "00" is input here, then the axis is not monitored. If an auxiliary input number has been specified and if there is no signal at the specified input, no positioning will take place. The DLC processes all blocks not containing any position commands. As soon as the program processing comes to a block containing a position command, the DLC will stop in this block until there is a signal at the input. If the specified auxiliary input signal shuts off during a positioning, the axis will be stopped and the error message "Feed Angle Loss 1" will be displayed.

2. Feed Interrupt or Feed Hold (Active Low)

This parameter specifies whether an interruption of the programmed feed or position command currently being process is possible or not. If "00" is input here, there will be no monitoring for axis interruption. If an input number is assigned, and an interrupt is present, the axis positioning command will not be processed. The DLC will continue to process all blocks not containing any position commands, regardless of how A120 is set. As soon as the program processing comes to a block containing a position command, the DLC will stop in this block until there is a signal at the input. If a position command is currently being processed and the interrupt is present, the axis will decelerate to a stop.

WARNING: The continuation of the axis positioning will occur immediately as the auxiliary input, Feed Interrupt returns high. Also, if the feed angle monitoring has been specified for the axis, the Feed Interrupt input number must be different (the two functions cannot use the same input number).

3. Axis Stop

This output becomes ‘1’ as soon as the following two conditions are met:

Command Position Value = 0

Actual Value-Per-Minute < 6 UpM

For functions that directly affect the position control circuit (e.g. following axis, SIN command), this output does not apply.

Parameter A121 - Maximum Motor Speed

```
Max. Speed [RPM]

A121
2000 0000
```

2000 - Maximum motor RPM

0000 - Unassigned set to 0000

This parameter defines the maximum RPM of the motor.

Example: A121 entry for 2000 RPM motor is:

```
Max. Speed [RPM]

A121
2000 0000
```

Parameter A122 - Monitor Window

```
Monitoring

A122
0 0 00 0 010
```

0 - Error Reports

0 = Error reports “Drive Runaway,” “Excessive Position Lag” or “Encoder Error” active

8 = Error reports deactivated

0 - Monitor window delay in delay x 100 milliseconds (0=OFF, 1-9 = delay)

00 - This output is set if the positional deviation is greater than the allowable deviation. An entry of 00 = no output programmed

---

Note: The output that is programmed here is only set if the error reports have been deactivated and the error occurs. Deactivation by entering "8" in the first screen position of this parameter or by setting flag 95 in the program.

---

0 - Unassigned

010 - Allowable position deviation in percent. In the normal case a value of 10-20% is input here.

Entry range: 001-100

Output: Position error > Maximum allowable deviation.

The DLC1 continuously monitors position control. To do this, the DLC uses a mathematical model to recognize an error immediately. “Drive Runaway” occurs if the current position of the encoder exceeds the target position of the model. “Excessive Position Lag” occur if the current position of the encoder is less than the position expected by the model.

The maximum allowable deviation between the actual position and the position calculated in the model is specified in % in parameter A122.

Program the maximum permissible deviation (in percentage) between the actual position and the position calculated by the model, in this parameter. For the monitoring to function properly, program a value in this parameter greater than the percentage of deviation that results under normal operation. Otherwise, nuisance faults will occur.

Calculate the maximum positional deviation:

$$\text{Position Deviation in IU} = \frac{\text{Max. speed (IU/sec)}}{\text{Position Gain (KV)}} \times \frac{60}{1000}$$

$$\text{Max.} \pm \text{Position Deviation in IU} = \frac{\text{RPM (as in A121)} \times \text{VK (as in A108)}}{\text{Kv factor (as in A103)} \times 1000}$$

Example:

max. RPM (A121)	=	2000 RPM
IU VK (A108)	=	0.39 in
Kv (A103)	=	1
Monitoring (A122)	=	20%

$$\text{Max.} \pm \text{Position Deviation in IU} = \frac{2000 \times 0.39\,\text{in.}}{1 \times 1000} = 0.78\,\text{in.} = 100\%$$

20% corresponds to 40 mm

If A122 = 00000020, the DLC will issue an error if the actual following error differs from the expected following error by more than 20%. Take 20% of 0.78 in: 0.20 x 0.78 = 0.16. At a positional deviation of 0.16 in., the controller interrupts the positioning process and reports a Excessive Position Lag or Drive Runaway.

A Drive Runaway would occur if the error exceeded 0.78 + 0.016 = 0.796 inches. Excessive Position Lag would occur if the following error went below 0.78 - 0.016 = 0.764 inches. When feeding at a lower speed the following error will be lower, but the maximum deviation from the expected following error will still be ± 0.16 inches.

Note the following:

- If you change the parameters Position Gain or Maximum Velocity, you do not need to reprogram the Monitoring Window, because it is entered in percentage of maximum position deviation.
- If the Monitoring Window entry, as converted, is smaller than the Position Tolerance, the DLC is unable to correctly distinguish between "Drive Runaway" and "Excessive Position Lag." Therefore, the converted Monitoring Window parameter should be greater than the Position Tolerance.
- If the programmed entry for Monitoring Window is too small, even a normal feed will generate a "Drive Runaway" or "Excessive Position Lag."
- The Monitoring Window value depends on your application. You should enter the lowest percentage possible that will not cause nuisance faults.

Possible causes for "Drive Runaway"

- The axis moved when no command value has been output.
- The maximum RPM in A121 is smaller than specified RPM, check the DSM software module.
- A122 entry is too small *(typical setting is 10%)*.

Possible causes for "Excessive Position Lag"

- Position command issued, but no movement detected:
- The motor cannot turn because of a mechanical bind.
- The maximum RPM in A121 is greater than specified RPM, check the DSM software module.
- The acceleration is too great (A102, also A117 second acceleration)

Parameter A123 - Follow Axis / Measuring Wheel Mode (Requires DEF Card)

Note: This parameter's function is changed as of software version DA01.1-01.2!

```
Followaxis

A123
4 0 0 0 00 00
```

4 = Mode

0 = Normal Feed Axis

2 = Follow axis (only DEF card)

3 = Measuring wheel mode, Version 1 (Control with measuring encoder)

4 = Measuring wheel mode, Version 2 (Measuring encoder as feed correction) (only DEF card)

5 = Feed axis with external encoder (DEF card necessary)

6 = Feed axis controlled

0 = For follow axis: (only DEF card)

1 = Axis 1 follows external encoder 1 (parameters B016 to B019)

2 = Axis 1 follows external encoder 2 (parameters B020 to B023)

For measuring wheel mode: (only DEF card)

1 = Measuring wheel with external encoder 1 (Parameters B016 to B019)

2 = Measuring wheel with external encoder 2 (Parameters B020 to B023)

For feed axis with external encoder:

1 = External encoder 1 (Parameters B016 to B019)

2 = External encoder 2 (Parameters B020 to B023)

0 = Not Used

0 = Not Used

00 = For measuring wheel mode and feed axis with external encoder:

Monitoring window in percent (00 to 99) between motor encoder and external encoder

00 = Monitoring switched OFF

00 = For measuring wheel mode:

Input "Encoder active" Input 01 to 37

00 = Encoder always active

Note: The DEF Incremental Encoder Input Card must be installed in the Indramat Digital Drive to accept the Master Encoder / Measuring Wheel Input.

The Follow Axis, parameter A123, is used to configure the axis to follow a master encoder input or as a measuring wheel encoder to verify the axis has been positioned properly.

The Following Operating Modes available are:

Normal Feed Axis
00000000

Following Mode (Axis follows Master Encoder)
21000000

Measuring Wheel Version 1
3100XXXX

Measuring Wheel Version 2
4100XXXX

When parameter A123, Follow Axis, is programmed with all zeros, the axis is a normal feed axis.

When using the Follow Axis, Measuring Wheel Version 1 or 2, the following parameters must be programmed:

External Encoder 1	Parameter B016
External Encoder 1, Lines/Rev	Parameter B017
External Encoder 1, Constant	Parameter B018
External Encoder 1, Offset	Parameter B019

When the Follow Axis, parameter A123, is programmed with 21000000, the axis will follow the master encoder input in Automatic or Manual Mode.

When the Follow Axis, parameter A123, is programmed with 3100XXXX, the Measuring Wheel Version 1 is selected. In Automatic Mode, the Measuring Wheel Version 1 will close the position loop on the measuring wheel encoder. Any slight difference between the load and motor position will be compensated for during the position move.

When the Follow Axis, parameter A123, is programmed with 4100XXXX, the Measuring Wheel Version 2 is selected. In Automatic Mode, the Measuring Wheel Version 2 will close the position loop on the axis. The axis position is compared to the measuring wheel encoder position and the difference is compensated for, during the axis move. When the axis is in the in-position tolerance window (defined in parameter A106, Position Tolerance), no compensation for the difference will occur. Compensation for the difference occurs when the axis is commanded out of the position tolerance window. This allows a degree of backlash in the mechanical system without effecting the stability of the servo. The following formula should be used when using this option to get optimum results.

$$\text{Position Tolerance} = \frac{\text{Acceleration rate}}{(\text{Gain} \times 16.66)^3}$$

When using the Measuring Wheel Version 1 (3100XXXX) or Version 2 (4100XXXX), the machine builder can program a percentage of deviation and an auxiliary input to enable the measuring wheel function. The percentage of deviation defines how much difference between the measuring wheel encoder position and the axis position will be allowed. When the difference becomes greater than this percentage the error "Encoder Fault A1" will occur.

Note: If "00" is specified for the percent deviation, there will be no comparison between the measuring wheel position and the motor's position.

Parameter A124 - Free

```
Free

A124
00000000
```

0000000- Not used, must be set to 0000000

Parameter A124 Free, is currently undefined.

Parameter A125 - Jerk Constant

(as of software version DA1-04VRS)

```
Jerk Constant

A125
0000 0.000
```

0000- Not used

0.000- Time constant for acceleration (s)

Range: 0.000 - 1.024

0 = Constant acceleration

The ‘Time constant for acceleration’ displays the time in which the DLC builds up the acceleration.

Parameter A126 - Position Window

(as of software version DA1-04VRS)

```
Position-Window

A126
00   123.456
```

00 - Output

123.456 - Position window

The DLC stores the last programmed move position. If the drive moves from the last programmed move position by ± the value listed in the position window, the programmed output will be turned off.

Parameter B000 - Enable Axis 2 (DG software only)

```
Enable Axis 2

B000
00000000
```

0000000= Not used, must be set to 0000000

Parameter B001 - System Inputs (DG software only)

```
System Inputs

B001
87654321
```

8= Clear external

7= Jog reverse

6= Jog forward

5= Immediate stop

4= Start

3= E-Stop message

2= Automatic

1= Parameter

For all of the system inputs listed above:

0 = Read in system input via DEA 4.1

1 = Read in system input via serial interface

2 = Read in system input via Interbus-S board DBS2.2 (DI software only)

Set outputs for DEA4 - DEA6 via the serial interface (or DBS2.1).

Read in inputs for DEA4 - DEA6 via the serial interface (or DBS2.1).

The assignment of the inputs and outputs is accomplished via parameters B001 and B002.

Parameter B001 - Serial Jog Enable (DA software only)

```
Serial Jog

B001
00000000
```

00000000 - Enable or disable Serial Jog

00000000 = Disabled

01100000 = Enabled

Parameter B002 - User I/O

(As of software version DA1-04V07/DG1-04V07)

```
User I/O

B002
00123456
```

00 = Not used

1 = Outputs A28 - A43 (DEA 6)

2 = Inputs E23 - E37 (DEA 6)

3 = Outputs A12 - A27 (DEA 5)

4 = Inputs E08 - E22 (DEA 5)

5 = Outputs A01 - A11 (DEA 4)

6 = Inputs E01 - E07 (DEA 4)

For all of the system inputs listed above:

0 = Read in inputs via DEA X.1

Outputs for DEA X.1 set by the DLC

1 = Read inputs via serial interface

Outputs for DEA X.1 set by serial interface

2 = Read in inputs via Interbus-S board DBS 2.2

Outputs for DEA X.1 set by Interbus-S board DBS 2.2 (DLI-software only)

a) The system outputs for the DEA 4 can be set only via the DLC.

b) The output map inside the DLC can differ from the actual status of the DEA 4 - DEA 6 display.

Note: If this parameter is redefined, it takes effect only after exiting and reentering Parameter Mode, or after restarting the DLC1.

Parameter B003 - Serial Interface

```
Serial Interface

B003
0960 0 1 8 1
```

The serial interface of the DLC can exchange data with peripherals in various ways. Define the transmission method in this parameter. Chapter 7 describes the interface in more detail.

If the two parameters B003 and B004 are stored in memory correctly, then the serial interface is ready for operation immediately. Incorrect entries in the parameters block the serial interface.

0960 - Baud Rate
- Min.= 300 Baud
- Max.= 19200 Baud
- 0000 = Interface not operating (to host computer)

Baud Rate input examples using common rates: 1920 = 19200 Baud
- 0960 = 9600 Baud
- 0480 = 4800 Baud
- 0240 = 2400 Baud
- 0000 = Interface deactivated

0 - Interface Operating Mode
- 0 = Standard RS232 (full duplex)
- 1 = Decade switch, IDS (2400 Baud, 8 data bits, 1 stop bit, no parity)
- 2 = Same as Mode 0
- 3 = Interface to SOT (Station Operator Terminal); RS232 or RS485 half duplex, one station only
- 4 = Serial bus link for SOT; RS485 half duplex, station 1 through 32 entered in parameter B004

Note: If you select the IDS option, do not enter the other values for this parameter. The DLC will configure the interface automatically.

---

1 - Parity
- 1 = No parity
- 2 = Even parity
- 3 = Odd parity

8 - Word Length
- 7 = 7 Bits
- 8 = 8 Bits

1 - Number of Stop Bits
- 1 = 1 Bit
- 2 = 2 Bits

Example: Decade switch query with IDS unit:

```
Interface
B003    00001000
```

0000 No meaning
1 Decade switch query
000 No meaning

The interface transmission parameter for the decade switch are set as follows: 2400 baud, 8 bit, 1 start bit, 1 stop bit, no parity check.

Parameter B004 - Serial Interface

```
Serial Interface

B004
1 1 08 9 T 00
```

1 - Checksum/Hardware Handshake (RTS/CTS)

0 = Checksum ON, RTS/CTS OFF

1 = Checksum OFF, RTS/CTS OFF

2 = Checksum ON, RTS/CTS ON

3 = Checksum OFF, RTS/CTS ON

4 = Disables checksum operation.

1 - Transmission Acknowledgment - " Y CR/LF "

0 = Acknowledgment OFF

1 = Acknowledgment ON

08 - Station Number, 01-32

Used with serial bus for SOT communication (See Parameter B003: Serial Interface - Interface Mode description)

9 - Error Code Over Serial Interface

0 = Function Disabled

1 = In case of DLC fault, automatically send an error message via the serial interface. See Status 53 in chapter 7 for details.

T = Serial Jog Time Out

Time = T*25msec, use T = 4 or 6 for best results.

00 = Transmission Delay For RS485 Communication (In Milliseconds)

00 = No Delay

02 = 2 msec (use for CTA10 or SOT)

This parameter allows the integration of the calculation of the check sum into the interface. The DLC will perform a bit count comparison from one operation (processing). If a "0" is not programmed in the first bit, a check sum is not required for communication, i.e. instead of "Checksum CR/LF," a "_CR/LF" is sufficient. If it is important that the data sent to the DLC is received correctly, it is recommended that the checksum be turned ON. The host device must be programmed to calculate and send the checksum with each transmission to use this option.

Transmission Acknowledgment, when active, returns a "Y" "CR/LF" from the DLC to the Host Device. This occurs after any command transmitted to the DLC that does not require any other response from the DLC.

The Transmission Delay For RS485 Mode, in milliseconds, is used to define a time delay for switching between the transmit and receive mode. In RS485 mode, when the last character is received, it switches immediately to transmit mode. This immediate switching can cause problems with some PC-RS485 driver cards, if they are not fast enough to switch to receive mode. The transmission delay allows the PC drivers to switch from transmit to receive mode without communication errors.

Parameter B005 - Memory Display

Note: For service use only!

```
Memory Display

B005
00 000000
```

This parameter is for use by the Indramat Service Department.

Set to 00 000000.

Parameter B006 - Start Task 2 & 3

```
Start Task 2 & 3

B006
1500 2000
```

1500 - Starting block number for second task

0001-2999 = Any Block number except 0000 or a block number of another task

0000 = Task 2 program disabled

2000 - Starting block number for third task

0001-2999 = Any block number except 0000 or a block number of another task

0000 = Task 3 program disabled

Task 1 always starts at block 0000.

This parameter allows running separate programs at the same time. As soon as the DLC is switched to Manual or Automatic mode, Task 3 begins running. As soon as Cycle Start is activated, in Auto Mode, Task 1 and 2 start running. Also, the system output, Automatic Cycle Running (Connector X17, pin 20) will turn on.

Note: Do not access the same routine by two tasks at the same time (within 1 millisecond) or a system fault will occur. The user program must be in place for task 3 before leaving the parameter mode because task 3 will start running immediately. If the task 3 program is not in place, you will see “ERROR INVALID BLOCK #.”

Warning: Do not use task 3 for servo commands. Task 3 continues running in Manual Mode and during an E-Stop.

Parameter B007 - Display Language / Decimal Place / Keypad Lockout

```
Language

B007
01 03 0 0 45
```

Use this parameter to specify in which language text will display on the CTA display and to set the number of decimal places used in positioning commands and certain parameters. It also allows disabling any program entry/changes from the CTA keypad.

01 - Language

00 = German

01 = English

02 = French

03 = Spanish

04 = Italian

05 = Portuguese

03 - Number of decimal places used in positioning commands and certain parameters

02 = Metric (e. g., millimeters, XXX.YY)

03 = US System (e. g., inches, XX.YYY)

The number entered for decimal places specifies how many digits are to the right of the decimal point for positioning commands and many parameters. Make your selection depending on the resolution required. See Appendix E, parameter input sheets, showing the decimal location for each parameter, with resolution set at 2 or 3 in B007.

Parameter # affected	**4th digit of B007**	**=2**	**=3**
A102, A107, A117		0	1
A100, A101, A106, A111, A113, A114		2	3
A108, B018		4	5
Command affected	**4th digit of B007**	**=2**	**=3**
POI, PSI, POA, PSA, PSM, POM, PST, VCC, VCA, BPT, BZP, PFA, PFI, REP, SAC		2	3

Figure 4-1: Decimal place settings

0 - Unassigned, set to 0.

0 - Enable "fault code via outputs"

0 = No fault code via outputs

1 = Fault code via outputs, 01 to 08

2 = Fault code via outputs, 09 to 16

3 = Fault code via outputs, 17 to 24

4 = Fault code via outputs, 25 to 32

5 = Fault code via outputs, 33 to 40

6 = Fault code via outputs, 41 to 48

If the function is enabled, an appropriate fault code is outputted via the programmed output when there is a fault. The fault code consists of 2 hex digits.

Fault code assignment:

00 to 3F - General disturbance

40 to 6F - Disturbance axis 1

For a complete list of fault codes and explanations, see chapter 8, Diagnostics and Troubleshooting.

45 - Lock-out feature

00 = Allows program entry/storage from the CTA keypad.

45 = Inhibits storage in memory of program blocks which are entered via the CTA keypad.

55 = Command data, but not commands, can be altered (as of software version DA1-04VRS).

Note: Programming can still be accomplished via the RS232/485 serial interface, such as by downloading from MotionManager program development tool. The lock-out feature prevents accidental or unauthorized changes via the CTA keypad.

Parameter B008 - M Function Inputs and Outputs

(DLC-G only)

```
M Function: I/O

B008
0000aaee
```

0000 - Not used, set to 0000

aa - First M function output; ascending

ee - First acknowledgment input; ascending

If no input is programmed (00), then the call-up of an M function with input acknowledgment in the user program leads to the error message ***“M= = Command MO/1.”***

If no output is programmed (00), then the call-up of an M function in the user program leads to the error message ***“M= = Command.”***

The input and output specified in this parameter become the starting point for the user defined M function I/O.

Example: If Aux. Output #4 is entered here as the First M-Function output, then output #5 becomes the second M function output, etc. These input/output designation changes remain true for the I/O sets configured for high and low bytes in the M function parameters.

See Section 4.6 for a complete description of M-Functions and how they are used.

Parameter B009 - M Function Timer (DLC-G only)

```
M Function Timer

B009
aa00tttt
```

aa = Output used to signal “Timer is running”

00 = Not used, set to 00

tttt = Timer for M function. Timer is entered in seconds between 0.000 to 9.999

This timer controls the ON time for each M-Function output turned on.

See Section 4.6 for a complete description of M-Functions and how they are used.

Parameter B010 - Free

```
Free

B010
00000000
```

0000000 - Not used, must be set to 0000000

Parameter B010 Free, is currently undefined.

Parameter B011 - Manual Vector

```
Manual - Vector

B011
07 1 0 2500
```

07 = Aux. input number to initiate Manual Vector program start with rising signal edge

00 = Manual Vector Disabled

01-37 = Aux. Input Number

1 = 0 = Start only with aux. input number

1 = Start with either aux. input number or any time you switch into manual mode.

0 = Not used, set to 0.

2500= Start block of Manual Vector program, Block 0000-2999.

Note: You must use the Command RTS, Return from Subroutine, to terminate the Manual Vector program.

This function allows you to run a user program in Manual Mode. This program must conclude with an "RTS" (the subroutine stack will not be changed). This program must not contain any feed instructions. The program is aborted if there is a switch over from "Manual" to "Automatic" or to "Parameter Mode." The program is started externally by means of a rising signal edge at one of the aux. inputs. If "00" is input for the aux. input, the Manual Vector is disabled. When programming Parameter B011, make sure that the start block of the Manual Vector program is not located in the main program.

Notes: The manual vector input is not accepted during jogging or homing (in the manual mode).

While the manual vector program is running, jogging or homing is not possible.

It is not possible to start the manual vector program unless the axis is enabled.

Parameter B012 - Program Interrupt Vector, Jump on Event

```
Interruptvector

B012
08 1 2 1650
```

For immediate interruption of main program (Task 1 only), start Interrupt Vector program. Can be used as an "Emergency Return" procedure.

08 -	00	= Disables Interrupt Vector
	01-37	= Aux. input number used to initiate Interrupt Vector program on rising edge of signal
1 -	0	= Delays the function while main program is in a sub-routine (JSR), waits for sub-routine to finish before continuing to the Interrupt Vector program (rising edge)
	1	= Means interrupt anytime, even if in a subroutine (Interrupt Vector always active [rising edge])
	2	= Same as 0, except falling edge
	3	= Same as 1, except falling edge
2 -	0	= A started position command will finish before executing the Interrupt Vector program
	1	= Position command is interrupted (drive(s) braked until stopped) and execute the Interrupt Vector program

1650-Start block of Interrupt Vector program, Block 0000-2999

The Interrupt Vector permits you to interrupt a user program externally at any time. The program sequence will then continue at the start block number specified for the Interrupt Vector program. No RTS command is required at the end of the Interrupt Vector Program. There is no return jump to the interrupted main program.

Notes: When a main program is interrupted, it cannot be resumed. It must be started over.

Interrupt vector only interrupts Task 1. Task 2 & 3 continue to run.

The Interrupt Vector can be called up only in Automatic operation. The "Cycle Start" and "Cycle Stop" remain effective.

The sub-program stack (JSR, RTS) is cleared each time the Interrupt Vector program is called up. A call to the Interrupt Vector while a sub-program is running, "xx0xxxxx" is stored in B012 until all sub-programs have been processed. Only then will the program continue with the "Interrupt Vector" program.

Parameter B013 - Velocity Override

(as of software version DLC1.1-DA1-04V00)

```
Override

B013
5 0 000000
```

5 - Axis Velocity Override

0 = Override off

4 = Override per Interbus S-object (axis 1 only)

5 = Override binary-encoded through inputs 01-07

6 = Override gray-coded inputs 04-07

7 = Override binary coded over 15 inputs (8-22), as of software version 04V08

0 - Override Axis 2

000000 - Unused, set to 000000

The Velocity Override parameter allows the machine builder to change the velocity anytime. The velocity can be changed by a binary input using auxiliary inputs 1 through 7 or by gray code inputs using auxiliary inputs 4 through 7.

Gray Code Override - The following input assignments result in the noted speed. Note the velocity in the right column, enter the respective number on that line for the input number (top line).

Input Number	**04**	**05**	**06**	**07**	**08**
Significance of Acknowledgement	2^0	2^1	2^2	2^3	Speed in %
	0	0	0	0	0
	1	0	0	0	1
	1	1	0	0	2
	0	1	0	0	4
	0	1	1	0	6
	1	1	1	0	8
	1	0	1	0	10
	0	0	1	0	20
	0	0	1	1	30
	1	0	1	1	40
	1	1	1	1	50
	0	1	1	1	60
	0	1	0	1	70
	1	1	0	1	80
	1	0	0	1	90
	0	0	0	1	100

Table 4-2: Input assignments

Parameter B014 - Restart Vector

(as of software version DA1.1-03.03/DG01.1-03.03)

```
Re-Start Vector

B014
0210 2700
```

02 - Input ‘Begin restart’

00 = Restart function not enabled

00 - Output ‘Restart possible’

00 = No output

2700 - Starting block for the Restart Vector program (0001-2999)

0000 = No restart program set. Status will be reestablished immediately.

<>0000 = Starting block for the Restart Vector program. Complete status reestablished at end of restart program.

Note: Use this parameter only if the system is equipped with a multi-turn feedback.

This parameter is used to define the starting block for the Restart Vector routine. If a program is interrupted by a power loss, system error or mode change, the status of outputs, absolute target position and velocity are temporarily stored in memory.

Based on the type of error and system configuration, it may be possible to restore the status of the DLC as before the interrupt.

Note: The temporarily stored absolute target position will only be resume if the axis is equipped with an absolute encoder.

Parameter B015 - Cycle Time

(as of software version DA01.1-03V05 / DG01.1-03V05)

```
Cycle-Time

B015
0 00000 00
```

0 - Number of program blocks per cycle time for Task 3

Default = 0 for 1st block/cycle

00000 - Not used

00 - DLC Position Control Cycle Time:

00 = default

10 = 1 ms

20 = 2 ms

Note: For all other values above, the DLC automatically sets the cycle time to the following default values:
1 axis programmed : 1 ms
2 axes programmed : 2 ms

Parameter B016 - External Encoder 1: Encoder Selection

(as of software version DA01.1-01.2)

Note: Requires DEF card!

```
External Encod. 1

B016
1 0 1 00000
```

1 = System selection of hardware

1 = DEF 1.1 card

2 = DEF 2.1 card

3 = DFF card (as of software version DA01.1-01.8)

0 = Not Used

1 = Operating direction of the encoder

0 = Encoder shaft and motor shaft have **same** rotational direction

1 = Encoder and motor shaft have **opposing** rotational direction

00000 = Not Used, set to 00000

Note: The DEF card can also be used for the functions “Follow Axis” and “Measuring Wheel Mode.” These functions are activated in parameter A123.

Parameter B017 - External Encoder 1: Pulses/Revolution

(as of software version DA01.1-01.2)

Note: Requires DEF card!

```
Ext.E.1  Impulse

B017
000 05000
```

Note: For incremental encoder only. (Requires DEF card.)

000 = Not Used

05000 = Encoder line count (pulses per encoder revolution) (100 to 10000)
Enter "00000000" for DFF card

Parameter B018 - External Encoder 1: Feed Constant

(as of software version DA01.1-01.2)

Note: Requires DEF or DFF card!

```
Ext.E.1 Constant

B018
020.00000
```

Feed constant of external encoder 1 in input units.

The distance per encoder revolution.

Parameter B019 - Offset Dimension External Encoder 1

Note: Requires DEF card!

```
Ext.E.1   Offset

B019
00000000
```

00000000 - Not used, must be set to 00000000

Parameter B020 - External Encoder 2: Encoder Selection

(as of software version DA01.1-01.2)

Note: Requires DEF card!

```
External Encod. 2

B020
1 0 0 00000
```

1 = System selection of hardware

1 = DEF1.1 card

2 = DEF2.1 card

0 = Not Used

0 = Operating direction of the encoder

0 = Encoder shaft and motor shaft have **same** rotational direction

1 = Encoder and motor shaft have **opposing** rotational direction

00000= Not Used, set to 00000

Note: The measuring wheel encoder can also be used for the functions “Follow Axis,” “Measuring Wheel Mode” and “Positioning with External Encoder.” When two external encoders are used simultaneously the system selection in parameters B016 and B020 must not be identical! These functions are activated in parameter A123.

Parameter B021 - External Encoder 2: Pulses/Revolution

(as of software version DA01.1-01.2)

```
Ext.E.2  Impulse

B021
000 05000
```

Note: For incremental encoder only. (Requires DEF card)

000 = Not Used

05000 = Encoder line count (pulses per encoder revolution) (100 to 10000)

Parameter B022 - External Encoder 2: Feed Constant

(as of software version DA01.1-01.2)

Note: Requires DEF card!

```
Ext.Enc.2    VK

B022
050.00000
```

Feed constant of external encoder 2 in input units.

The distance per encoder revolution.

Entry range: From 000.0,1000 to 500.0,0000 in input units

Entry resolution: 4 or 5 digits after the decimal point (see parameter B007)
(Also see parameter A108)

Parameter B023 - Offset Dimension External Encoder 2

Note: Requires DEF card!

```
Ext.Enc.2 Offset

B023
00000000
```

00000000 - Not used, must be set to 00000000

This parameter is currently under development.

Parameter C000 - Analog Output: Channel 1

(as of software version DA01.1-01.6)

```
Analog Output 1

C000    00000002
```

0000000 - Not used, must be set to 0000000

2 - Output type

0 = Current command

1 = RPM command

2 = RPM actual value from feedback

3 = Position - Actual value from feedback

4 = Motor encoder: Sinusoidal feedback

5 = Motor encoder: Cosinusoidal feedback

Parameter C001 - Analog Output: Channel 2

(as of software version DA01.1-01.6)

```
Analog Output 2

C001    00000005
```

0000000	-	Not used, must be set to 0000000
5	-	Output type

0 = Current command
1 = RPM command
2 = RPM actual value from feedback
3 = Position - Actual value from feedback
4 = Motor encoder: Sinusoidal feedback
5 = Motor encoder: Cosinusoidal feedback
6 = DLC Output (e.g. Axis 2)

Parameter C002 - Overload Factor

(as of software version DA01.1-01.6)

```
Overload Factor

C002    00000400
```

00000 - Not used, must be set to 00000

400 - Overload factor in percent (000-400)

Parameter C003 - Position Data Scaling at Analog Outputs AK1 and AK2

(as of software version DA01.1-01.6)

```
Position Data
Scaling A-Output
C003   0000360.0
```

0000360.0 - Degrees/10V (0000000.1 -1474560.0)

Parameter C004 - Velocity Data Scaling at Analog Outputs AK1 and AK2

(as of software version DA01.1-01.6)

```
Velocity Data
Scaling A-Output
C004    00002000
```

000 - Not used, must be set to 000

02000 - Degrees/10V (00002 - 65000)

Parameter C005 - Velocity Loop Monitoring

(as of software version DA01.1-01.6)

```
Velocity Loop
Monitoring
C005    00000001
```

0000000 - Not used, must be set to 0000000

1 - Monitoring Circuit for Velocity Loop

0 = OFF

1 = ON

Parameter C006 - Position Velocity Window

(as of software version DA01.1-01.6)

```
Position
Velocity window
C006   0005.0000
```

0005.0000 - RPM (0000.0000 - 0005.0000)

Parameter C006 is only effective in conjunction with a multiturn encoder. When using a single-turn encoder, C006 is overwritten with 0 after the drive is powered up again.

Parameter C007 - Absolute Encoder - Reference Position

(as of software version DA01.1-01.6)

```
Absolute Encoder
Ref. Position
C007   0000.0000
```

0000.0000 - Degrees (0000.0000 - 0359.9000)

Parameter C007 is only effective in conjunction with a multiturn encoder. When using a single-turn encoder, C007 is overwritten with 0 after the drive is powered up again.

Parameter C008 - Error Reaction

(as of software version DA01.1-01.6)

If there is a drive fault during motion, this parameter determines the drive's error reaction to the fault.

```
Error Reaction

C008    00000002
```

0000000 - Not used, must be set to 0000000

2 - Error Reaction

0 = The drive switches itself to velocity control and defaults to a RPM setpoint of 0.

The brake engages after maximum 400 msec. and it disengages after another 100 msec.

DDS: Error report to the power supply (package reaction).

DDS: Error report to the internal power supply (Failure of power system protection.

1 = Same as error reaction 0, but no error report to the power supply (package reaction), or to the power supply.

2 = The drive disengages immediately. If there is a brake, it engages immediately. No error reaction (no package reaction).The drive switches itself to velocity control and defaults to a RPM setpoint of 0.

Parameter C009 - Current Loop Proportional Gain

(as of software version DA01.1-01.6)

```
Current Loop
P_Gain
C009   000120.00
```

000	=	Not used, must be set to 000		
120.00	=	Volts/Amp		
		Minimum	=	001.00-030.00
		Maximum	=	015.00-300.00

Parameter C010 - Velocity Loop Proportional Gain

(as of software version DA01.1-01.6)

```
Velocity Loop
P_Gain
C010    00000800
```

000 = Not used, must be set to 000

00800 = mAsec/rad (00000 - 65000)

Parameter C011 - Velocity Loop Integral Reaction Time
(as of software version DA01.1-01.6)

```
Velocity Loop
I_Reaction Time
C011   0002500.0
```

000 = Not used, must be set to 000

2500.0 = msec (0000.1 - 6500.0)

Parameter C012 - Smoothing Time Constant

(as of software version DA01.1-01.6)

```
Smoothing Time
Constant
C012    00000600
```

000 = Not used, must be set to 000

00600 = µs (00250 - 65000)

Parameter C013 - External Brake Release

```
External Brake
Release
C013    00000001
```

0000000 = Not used

1 = Enable “External Brake Release” function in the drive

Parameter C100 - Set Standard Drive Tuning Parameters Via CTA

(as of software version DA01.1-01.6)

From the C-Parameter group, pressing the blank key again displays the C100 parameter. In this menu, the drive parameters can be set to their default values.

```
Set Standard Drv
Params Over CTA
C100    00000000
CODE:   00005301
```

This menu consists of only the parameter C100.

The standard parameters are set after entering the number 00005301 and pressing the block store key.

After pressing the block store key this menu item exits to “Drive Parameter C000.”

If the entry is made incorrectly, the menu exits to “Drive Parameters” without setting the standard parameters, i.e. there is no control monitoring of the entry.

Parameter C101 - Set Absolute Position

(as of software version DA1-04 / DG1-04V04)

```
Set Absolute
Position
C101   +00000.00
```

Input of the Absolute Reference Position in input units.

Currently available only if Homing Setup is programmed (Parameter A110).

When Homing Setup is active, setting the absolute encoder reference position via the drive parameter is problematic.

The new DLC Parameter 101 offers a significantly simpler solution by setting the position in input units.

4.6 M Functions

General Description

Pre-defined M functions are available to simplify handshaking between the DLC and the line control during program execution. Pre-defined means that the given M function affects fixed assigned outputs that are to be acknowledged by inputs that are also fixed.

Any inconsistency in the acknowledgments leads to the program stopping in Task 1 and 2, as well as axis movement halting, until restoration of the defined acknowledgment. Up to 64 predefined M functions can be used. Outputs and acknowledgment inputs should always be sequential. M functions are monitored during processing of the selected command or if programmed as such, they can be permanently monitored.

Selection occurs in the running program (Task 1 or 2) in the GO1 command or by the M word as a separate command.

```
E 1000 G01
1 ±000010.00 mm
```

M== Selection of an M Function

```
E 1000 M==
mm
```

mm - Number (00...63) of the M function to be selected

A pre-coded M function is either set in a separate block using “M== “ or in combination with a G01 command. If the M function is programmed in a G01 block, it is set after the programmed position has been reached (within the limits set by A106, Position Tolerance). If G01 is programmed within a G61 contouring mode, the M function is set so as to allow optimum speed.

If the M functions set are time dependent, it is important to note that their status will change according to the M function table after the set time in Parameter B009 has expired.

The status of the inputs and outputs, after processing the M function, are defined in the M Function table. The table of M functions is specified in parameters only and can be modified by the CTA keyboard or seriallly through the RS 232/RS 485 port.

M Function Table Set-Up

To enter M Function data, you must be in Parameter Mode. After entering Parameter Mode, use the up or down arrow keys until your display looks like one of the two shown below.

The M Function table has the following format for every pre-defined M function:

```
M Functions

MHmm  xxxxxxxx
    High Byte
```

```
M Functions

MLmm  xxxxxxxx
     Low Byte
```

mm - Number of the M function (00...63)

Sixteen inputs/outputs can be set with every M function.

```
M Functions

MLmm  8xxxxxx1
```

1 - 1st output/1st input
8 - 8th output/8th input

```
M Functions

MHmm  16xxxxx9
```

9 - 9th output/9th input
16- 16th output/16th input

The first user output and the first user acknowledgment input are defined in parameter B008. Every position "x" corresponds to an input/output pair and can assume the following significance:

x = 0 Turn output OFF with input acknowledgment
x = 1 Turn output ON with input acknowledgment
x = 2 Output unchanged with no acknowledgment
x = 3 Turn output OFF with no input acknowledgment
x = 4 Turn output ON with no input acknowledgment
x = 5 Turn output ON with no acknowledgment; set timer
x > 5 Output unchanged with no acknowledgment
x = 0/1

After calling up the M function, it waits until all acknowledgment conditions have been satisfied. Then, stepping to the next program block and permanent monitoring of the acknowledgments occurs. As soon as one of the conditions is not satisfied, the program stops in Task 1 and Task 2, axis movements are stopped, and the message ***"Stop Active"*** is displayed. The program starts running again as soon as all acknowledgment conditions are satisfied.

X = 5

After the time defined in parameter B009 has elapsed, all bits that have been turned ON will be turned OFF. Since there is only one timer, a new M function using the time function cannot be set until the timer has run out. In this case, a timer that has not yet run out has a limiting effect on the processing of any subsequent M Functions.

4.7 Parameter Data Sheets

A Parameter Input Sheet

MAXIMUM VELOCITY* A100 0 _ _ _ _ _ _ _	MAXIMUM TRAVEL A114 +_ _ _ _ _ _ _ _
JOG VELOCITY A101 0 _ _ _ _ _ _ _	SPECIAL FUNCTION A115 _ _ _ _ _ _ 0 _
ACCEL RATE* A102 _ _ _ _ _ _ _ _	ROTARY TABLE A116 _ _ _ _ _ _ _ _
POSITION GAIN* A103 0 0 0 0 _ _ _ _	KNEE POINT A117 _ _ _ _ _ _ _ _
ENCODER A104 _ _ 0 0 0 0 0 0	BRAKE RELEASE A118 _ _ _ _ 0 0 0 0
FREE A105 0 0 0 0 0 0 0 0	FREE A119 0 0 0 0 0 0 0 0
POSITION TOLERANCE A106 _ _ _ _ _ _ _ _	FEED ANGLE MONITORING A120 _ _ _ _ _ _ 0 0
POSITION PRE-SIGNAL A107 _ _ _ _ _ _ _ _	MAXIMUM SPEED (RPM)* A121 _ _ _ _ 0 0 0 0
FEED CONSTANT* A108 _ _ _ _ _ _ _ _	MONITOR WINDOW A122 _ _ _ _ 0 _ _ _
DIRECTION A109 0 _ 0 0 0 0 0 0	FOLLOW AXIS A123 _ _ 0 0 _ _ _ _
HOMING SETUP A110 0 _ 0 _ _ _ 0 _	FREE A124 0 0 0 0 0 0 0 0
HOMING OFFSET A111 _ _ _ _ _ _ _ _	JERK CONSTANT A125 0 0 0 0 _ _ _ _
HOMING ACKNOWLEDGMENT A112 _ _ _ _ 0 0 _ _	POSITION WINDOW A126 _ _ _ _ _ _ _ _
MINIMUM TRAVEL A113 ±_ _ _ _ _ _ _ _	
* may be changed only with permission from the machine/drive manufacturer	

Notes:

1. To get to the other parameter sets, either press the Store key or use the up and down arrow keys.
2. To increment or decrement through the parameters in each set, use the + or - keys.
3. To move the cursor in the data field, use the right or left arrow keys.

B Parameter Input Sheet

ENABLE AXIS 2 B000 _ 0 0 0 _ _ _ _	INTERRUPT VECTOR B012 _ _ _ _ _ _ _ _
SYSTEM INPUT B001 _ _ _ _ _ _ _ _	OVERRIDE B013 _ _ 0 0 0 0 0 0
USER I/O B002 0 0 _ _ _ _ _ _	RESTART VECTOR B014 _ _ _ _ _ _ _ _
SERIAL INTERFACE B003 _ _ _ _ _ _ _ _	CYCLE TIME B015 _ 0 0 0 0 0 _ _
SERIAL INTERFACE B004 _ _ _ _ _ _ _ _	EXTERNAL ENCODER 1 B016 _ 0 _0 0 0 0 0
MEMORY DISPLAY B005 0 0 0 0 0 0 0 0	EXTERNAL ENCODER 1 LINES/REV B017 0 0 0 _ _ _ _ _
START TASK 2 & 3 B006 _ _ _ _ _ _ _ _	EXTERNAL ENCODER 1 FEED CONSTANT B018 _ _ _ _ _ _ _ _
LANGUAGE B007 _ _ _ _ 0 _ _ _	EXTERNAL ENCODER 1 OFFSET B019 _ _ _ _ _ _ _ _
M-FUNCTION: I/O B008 0 0 0 0 _ _ _ _	EXTERNAL ENCODER 2 B020 _ 0 _ 0 0 0 0 0
M-FUNCTION: TIMER B009 _ _ 0 0 _ _ _ _	EXTERNAL ENCODER 2 LINES/REV B021 0 0 0 _ _ _ _ _
FREE B010 0 0 0 0 0 0 0 0	EXTERNAL ENCODER 2 FEED CONSTANT B022 _ _ _ _ _ _ _ _
MANUAL VECTOR B011 _ _ _ 0 0 _ _ _	EXTERNAL ENCODER 2 OFFSET B023 _ _ _ _ _ _ _ _

Notes:

1. To get to the other parameter sets, press the Store key or use the up and down arrow keys.
2. To increment or decrement through the parameters in each set, use the + or - keys.
3. To move the cursor in the data field, use the right or left arrow keys.

C Parameter Input Sheet

ANALOG OUTPUT 1 C000 0 0 0 0 0 0 0 _	ERROR REACTION C008 0 0 0 0 0 0 0 _
ANALOG OUTPUT 2 C001 0 0 0 0 0 0 0 _	CURRENT LOOP P-GAIN C009 0 0 0 _ _ _ _ _
OVERLOAD FACTOR C002 0 0 0 0 0 0 _ _	VELOCITY LOOP P-GAIN C010 0 0 0 _ _ _ _ _
POSITION DATA SCALING ANALOG OUTPUT C003 _ _ _ _ _ _ _ _	VELOCITY LOOP I-REACTION TIME C011 0 0 0 _ _ _ _ _
VELOCITY DATA SCALING ANALOG OUTPUT C004 0 0 0 _ _ _ _ _	SMOOTHING TIME CONSTANT C012 0 0 0 _ _ _ _ _
VELOCITY LOOP MONITORING C005 0 0 0 0 0 0 0 _	EXTERNAL BRAKE RELEASE C013 _ _ _ _ _ _ _ _
POSITION VELOCITY WINDOW C006 _ _ _ _ _ _ _ _	SET TO STANDARD PARAMETERS C100 _ _ _ _ _ _ _ _
ABSOLUTE ENCODER REFERENCE POSITION C007 _ _ _ _ _ _ _ _	SET ABSOLUTE POSITION C101 _ _ _ _ _ _ _ _

Notes:

1. To get to the other parameter sets, press the Store key or use the up and down arrow keys.
2. To increment or decrement through the parameters in each set, use the + or - keys.

To move the cursor in the data field, use the right or left arrow keys.

5 Programming

The application program of the system is defined and entered by the user. It can be entered directly via the CTA keypad and display, or from a remote terminal device interfaced through the RS-232, or RS-485 port. The application program flow is similar to a Basic program. Three letter mnemonic commands are used. There are 3000 programming lines/blocks available for user programming, numbered 0000 through 2999.

The Indramat MotionManager^{TM} Program Development Tool provides an efficient method of creating and editing the user program for the DLC control. It is a software package that runs on any DOS-based computer. It provides several benefits over programming the DLC from its control panel. It also includes enhanced features for creating and editing program that are not possible from the DLC control panel. Refer to Publication IA 74733 for specific information on using this software program.

This chapter begins by describing some basic information that should be considered before creating a program for the DLC. It then describes the methods to enter the user program directly into the CTA keypad and display. It further describes the programming commands and their function in a user program.

5.1 Positioning

Two types of positioning can be selected in the system, absolute and incremental. All positioning is done in the units of your choice and are referred to as **Input Units (IU)**. Input Units are the user’s desired units of measure (i.e. inches, mm, radians, degrees, etc.).

In absolute positioning, all movements of the slide are made some absolute distance from the machine reference point. Thus, if the slide is at +2 inches from home, and an absolute position command to move +3 inches is executed, a one inch feed in the positive direction will result.

In incremental positioning, all movements of the slide are made in the commanded direction to the distance specified, starting from the current position. Thus, if a slide is at +2 inches from home, an incremental command to move +3 inches will result in the slide being positioned +5 inches from home.

5.2 Auxiliary Inputs/Outputs

The DLC has a set of I/O points with a predefined use. It also includes a set of I/O points which can be defined by the user for controlling machine functions. They will be referred to as auxiliary inputs and auxiliary outputs (or aux. inputs and aux. outputs). Auxiliary inputs are also known as acknowledgments. Certain commands are provided for use to address these inputs and outputs. The I/O Commands are described in section 5.8.5.

Warning: Auxiliary input and output numbers that have been dedicated by the machine builder for a specific purpose must not be changed. Personal injury or damage to the machine/drive train could result from such changes.

Programming Inputs/Outputs

Certain outputs are predefined in the DLC internal program and cannot be changed by the user program. When the DLC is powered ON, it sets certain outputs, per the internal program, to default position. Likewise, when a fault occurs, it sets many outputs OFF. Outputs are re-established, either through hardware or software, i.e. the Automatic Mode Indicator turns ON after that mode is selected by input (assuming all other conditions are met), a software flag is turned ON or OFF as the user program executes the block containing the command.

Chapter 3 describes the functional use of each system input/output, as well as many programming and parameter entries specifying input or output connections. Several I/O are available for use as flags in the user program. Certain output flags are set in firmware and can be queried by the user program. Table 5.1 list the hardware outputs that can be used in the program to electrically signal an external component. It also lists the output software flags that can be used internally in the program. It defines the output flags which are set in firmware. Refer to this table when programming an output.

Inputs/Outputs Signal Definition

There are two states that system inputs/outputs and auxiliary inputs/outputs could hold. The "ON" or High state, means that there is a +24 Vdc signal present at the input/output. The "OFF" or Low state, means that there is a O Vdc signal at the input/output. A signal line is described as “Active High” when its associated action is initiated by a High (+24 volts) signal level. It is described as “Active Low” when its function is initiated by a Low signal (0 volts). An active low signal must remain in the high state to allow normal operation. Refer to Chapter 3 for further description of I/O signals.

DLC with DEA 4	11	Hardware Outputs	1-11
	88	Software Flags	12-99
DLC with DEA 4 & DEA 5	27	Hardware Outputs	12-27
	48	Software Flags	28-99
DLC with DEA 4	43	Hardware Outputs	28-43
DEA 5, & DEA 6	80	Software Flags	44-99

Table 5-1: Output Definitions

Outputs and Flags	**1 to 72**	**are cleared (set to 0 volts) when:** • the DLC is first powered-up or if there is a loss of power • there is a system fault (hardware or program)
Output Flags	**73 to 80**	**are cleared (set to 0 volts) when:** • the DLC is first powered-up or there is a loss of power • an E-Stop error occurs, or the DLC is switched to Parameter Mode
Output Flags	**81 to 88**	**are retained in RAM (battery backed), they can only be cleared:** • if they are turned ON/OFF in the user program • if the battery is disconnected or fails
Output Flags	**89 to 94**	**are set in firmware and can be queried by the user program**
	89	"1" indicates Manual Mode
	90	"1" indicates Automatic Mode
	91	currently not used
	92	currently not used
	93	currently not used
	94	"0" indicates a system fault
Output Flags	**95 to 99**	**are set within the user program to provide specific actions; they are cleared when the DLC is first powered up or loses power**
	95	"x" Monitoring Window is turned OFF =1 or ON =0 (see Warning below)
	96	"x" currently not used
	97	"x" currently not used
	98	"1" axis 1 motion is interrupted (see Warning below)
	99	"x" currently not used

Warning: If the Monitoring Window is turned OFF, the DLC will have no way of detecting if a motor has Drive Runaway or Excessive Position Lag.

Warning: If motion is interrupted by setting flags 98 ON, it will resume automatically when the output is turned OFF.

5.3 Multi-Tasking

The DLC is capable of operating one motion program and two background PLC programs simultaneously when multiple tasks are programmed (see next section). This allows a single DLC to operate one motion program and two background processes at the same time or utilize multi-tasking in a single process.

5.4 Start of the Program

When the DLC is first powered up or an error is cleared, the program block pointer is set to block 0000 for Task 1. If Tasks 2 and 3 are used, they will start at their assigned starting point as user defined in parameter B006. All programs must start at their assigned starting points; Task 1 must start at block 0000. Blocks of numbers of programming will be followed sequentially unless a jump or branch instruction is encountered.

If block 2999 is executed and it is not a jump command, all motions are stopped, and an error code is displayed. Typically, the last command block you enter will be a jump command to return to the start of the next cycle or to return from a sub-program routine.

5.5 End of the Program

Take extreme care to control the flow of the program. This is especially important when using multi-tasking. Most programs are designed so they will loop back to the start of each task and wait for the proper sequence of events before starting again. Make sure that each task will not interfere with another task.

5.6 Programming Mode

The DLC must be in either Automatic or Manual mode to accept program entry/edit from the front panel. It is recommended that the DLC be placed in manual mode when editing the program, especially when this involves changing a command, or several blocks.

Warning: Program entry in Automatic Mode, while the unit is in operation, will be accepted as soon as the Store key is pressed. The next time the block is scanned in the program, the updated data will be executed. It is recommended that the DLC be placed in Manual Mode when editing the program. Complete and verify the program changes before returning to Automatic Mode. Have accurate listings of the program and parameters when editing, to reduce the possibility of errors.

5.7 General Format

The general format, shown on the CTA display during program entry / edit, is as follows:

```
E _0000 ABC
    (DATA)
```

E = shows the display is in the program edit mode

0000 = block number displayed; command or data can be viewed or edited.

You can select any block number 0000-2999 to display. To scroll through the block numbers, first press the **CR** key to locate the cursor in the top line (or use the left/right arrow keys). Then use the **+** and **-** keys to scroll up or down through the block numbers. You can also type the block number desired directly over the existing number.

ABC = 3-letter mnemonic of command programmed in displayed block

To scroll through the commands, position the cursor to the right of the displayed command three letter mnemonic. Use the up or down arrow keys to scroll alphabetically through commands. When the desired command is on the display, press the right arrow key to select the command and move the cursor into the data field that appears for the specific command on the second line.

(Data) = Most command requires entry of data specific for that command, as described in the following sections.

5.8 Command Summary

The various commands can be classified into categories by their function, as described in the following sections. These sections describe the general use and differences between all the programming commands. For future reference, you can use these sections to select the specific command desired for the general function. Then refer to that command description in section 5.9 for further details on how it works and requirements for its entry.

Positioning Commands

There are two major types of positioning commands: Incremental - movement from one point to the next point, or Absolute - movement in relationship to a known home point. These can be further broken into two different functional types of positional commands.

1. Commands that require the axis to be at the final programmed position, as determined by the In Position Signal, before stepping to the next block. These are identified by the letter "S" in the mnemonic (Stop).
2. Commands that, after being read into the position buffer, immediately go to the next block for further program execution. These are identified by the letter "O" in the mnemonic (Onward).

The following is a list of the position commands:

`G01` Linear Feed with M Functions

`G60` Position with Exact Stop

`G61` Begin Velocity Rate Optimization profile

`POI` Incremental position command

`PSI` Incremental position command with In Position Acknowledged

`POA` Absolute position command

`PSA` Absolute position command with In Position Acknowledged

`POM` Incremental or Absolute positioning to IDS thumbwheel switch setting

`PSM` Incremental or Absolute positioning to IDS with In Position Acknowledged

`PFA` Absolute position to a positive stop

`PFI` Incremental position to a positive stop

Position commands have the following characteristics:

- Position commands must include three components: axis, Position or Distance, and Velocity.
- Positions are programmed directly in units of your choice: inches, metric, degrees, etc.
- You can select either two or three decimal points of accuracy *(in parameter B007)*.
- The acceleration rates specified in parameter A102 will be used as defaults, unless they are changed with the ACC command prior to the motion.
- Select either Constant acceleration or Knee Point acceleration profile.
- Motion can be independent or simultaneous.
- Axis can be programmed for Rotary motion *(rotation resetting to zero after one revolution)*.

Position Support Commands

The following commands are used either in conjunction with a positioning command or by themselves to achieve the specific result.

`ACC` Change acceleration/deceleration rate (0.1 to 99.9% of A102)

`CLA` Clear absolute position buffer (set to zero)

`COC` Cam Output Control

`CON` Constant velocity command (sets axis to a continuous speed)

`CPL` Clear Position Lag

`D==` Tool Correction Memory Selection

`FAK` Factor all position distances by a set ratio

`FOL` Axis synchronization factor (master/slave mode)

`FUN` Functions

`F==` Feedrate command for G-Code Functions

`G40` Deactivate Tool Correction

`G43` Add Tool Correction from commanded position

`G44` Subtract Tool Correction from commanded position

`G74` Reference (home) axis

`G90` Begin Absolute Positioning

`G91` Begin Incremental Positioning

`HOM` Execute a homing routine

`KDI` Copy a position difference to a target block

`MOM` Torque Reduction

`PBK` Position break (axis immediate stop)

`PST` Position test (turn output ON/OFF based on position)

`REF` Move at a set velocity until a registration mark is detected

`REP` Maximum search distance for REF command (branch if exceeded)

`RMI` Registration mark interrupt (high speed)

`SAC` Set Absolute position reference

`SIN` Sinusoidal oscillation

`SO1` Read inputs to program a distance and velocity into memory

`VCA` Velocity change absolute

`VCC` Velocity change during a profile

`VEO` Velocity override (binary/gray code override)

`WRI` Write current position to a target block (Teach)

Branch Commands

These are commands for conditional program flow control, used to direct the actions of the control based on events that can be monitored by the DLC.

`BAC` Branch when the item count is met (up to 99,999 counts)

`BCA` Output-dependent conditional branch (1 to 99)

`BCB` Binary inputs conditional branch (16 or 256 possible targets)

`BCD` BCD input conditional branch (100 possible targets)

`BCE` Input-dependent conditional branch (1 to 37)

`BIC` Branch conditional on inputs (1 to 37)

`BIO` Branch if Input/Output mask matches (compares 10 bits)

`BMB` Binary output conditional branch (compares any # of outputs)

`BPA` Parallel output conditional branch (compares 10 outputs)

`BPE` Parallel input conditional branch (compares 10 outputs)

`BPT` Branch on position reached (must be at position)

`BZP` Branch if at or past a position

Jump Commands

Very similar to branch commands except they are not conditional; i.e., jumping takes place immediately when the command is read. Subroutines are also part of this category. They are common program sequences that are used repeatedly throughout a program, returning to the calling point when completed.

`APJ` Turn parallel outputs ON or OFF, then jump

`CST` Change subroutine stack level

`JMP` Unconditional jump to a block

`JSR` Jump to a subroutine

`JST` Unconditional jump to a block, then stop the program

`JTK` Immediately cause a jump to a block in a selected task (task interrupt)

`RTS` Return from subroutine

Auxiliary Functions

Refers to the monitoring of Inputs and controlling the states of Outputs. This ability is vital for a control to be capable of acknowledging and responding to the external environment and to control functions other than motion. Note that branch commands and certain position support and jump commands (PST, APJ) are also used for I/O interfacing.

Output Control Commands

`AEA` Turn a single output ON or OFF (1 to 88)
`APE` Turn a parallel group of output ON or OFF (10 outputs, group 0-9)
`APJ` Turn a parallel group of outputs ON or OFF, then jump (10 outputs)
`CIO` Copy a group of inputs or outputs to a set of outputs (10 outputs)
`COC` Turn 6 outputs ON or OFF based on Cam position
`M==` Select M-Functions
`PST` Position test (turn output ON/OFF based on position)
`STO` Send information to outputs

Input/Output Monitoring Commands

`AKN` Check a single input state, continue if state matches (1 to 37)
`AKP` Check a parallel group of inputs, continue if all match (10 inputs, group 0-9)
`ATS` Check a single output state, continue if state matches (1 to 99)
`M==` Select M-Functions

Counter Commands

Useful for tracking progress of a process. You can use any number of cycle/parts/batch counters within a program.

`BAC` Branch until the item count is met (up to 99,999 counts)
`CLC` Clears the counter at a block with a counter command in it (set to zero)
`COU` Turn output ON at end of count (up to 999,999 counts)

Timer Commands

The DLC firmware supports the delay of the program in a wait block for a preset time.

`WAI` Wait for a set time delay, then move to next block (10 ms to 99.99 sec)
`G04` Dwell Time (1 ms to 9.999 sec)

Other Commands

The following commands have the indicated function. Refer to the next section for a detailed description of every programming command.

`CID` Change instruction data
`NOP` No Operation
`RSV` Restart vector
`STH` Send to host (Communications)

5.9 Command Descriptions

This section describes each programming command. The first line of the illustrated displays are as previously described, with **E** indicating Edit Mode, a random block line number, the three letter mnemonic for the command and relative data fields.

This manual principally describes hardware versions DLC 1.1 and DLC 2.1 with software version DG01.1-XX.X. The software has parameter selected two or three decimal place precision in the commands that involve positioning- and position-related parameters (See Parameter B007).

Each command requires specific data. All places for required data in a user program block which contains illegal characters (a space or an undefined character - not a number or the +/- sign) are replaced by the asterisk character (*) in the display. This enables the user to easily recognize which points in a command data needs be programmed. A program block has the correct syntax when it contains no asterisk (*) characters in the display.

The following sections fully describe the use and data field entry requirements of each command.

ACC Acceleration Change

```
E  0001 ACC
1  750
```

1	-	Axis 1
750	-	Acceleration in percentage of the acceleration rate set in parameter A102
		Min.= 00.1%
		Max.= 99.9%
		One decimal place precision (i.e. 750 = 75.0%).

The ACC command allows the acceleration rate to be changed in the user program. The desired rate is programmed as a percentage of the acceleration entered in parameter A102.

Stepping to the next block takes place immediately after the block is read in. If you change this rate "on the fly" it will take effect starting with the next positioning command. A change in acceleration can only take affect if the feed comes to a complete stop. The change will not take affect if a feed is currently in progress.

The acceleration rate remains in effect for every positioning command until it is changed via another ACC command. If the DLC is taken out of Automatic Mode, the acceleration rate resets to the value programmed in the acceleration parameter A102.

Note: You can use the ACC command or Parameter A117 (Knee Point/Second Acceleration), but not both. When using one, the other does not function.

Example:

0000	NOP		; No operation
0001	ACC	1 750	; Change the acceleration rate to 75.0% of A102 (axis 1)
0002	PSI	1 +00100.000 999	; axis 1, incremental feed, with acknowledgment, of +100.000 IU* at max speed
0003	WAI	00.50	; Wait, allows for axis motion to completely stop
0004	ACC	1 999	; Change the acceleration back to max. (the value set in A102)
0005	JST	0001	; Jump and stop to block 0001
			* IU= Input Units- desired unit of measure for positioning

AEA Auxiliary Output ON/OFF

```
E  0020 AEA
07  0
```

07 - Auxiliary output number (01-99)

0 - Output state

0= turn output **Off**

1= turn output **On**

The AEA command is used to set the state of any single auxiliary output. Stepping to the next block occurs immediately after this program block is read.

The auxiliary outputs retain their status when the DLC exits Automatic Mode into Manual Mode.

In a fault condition or the entry to Parameter Mode, the auxiliary outputs will automatically be set to the Off (0) state. Upon the clearing of the fault or exiting of Parameter Mode, the auxiliary outputs remain in the Off state until their status is changed in the user program in Automatic Mode.

Warning: The manipulation of auxiliary outputs 89 through 99 can have unexpected results. See Table 5.1 for more information on auxiliary outputs that serve as status markers.

Example:

0000	JMP	0020	; Jump to block 0020
0020	AEA	07 0	; Turn auxiliary output 07 Off
0021	POI	1 00000.550 865	; axis 1, incremental feed of +0.550 IU at 86.5% of max velocity (A100)
0022	ATS	15 1	; Check auxiliary output 15 until On status (axis in position per tolerance set in A106)
0023	AEA	07 1	; Turn auxiliary output 07 On
0024	WAI	1.00	; Wait 1 second
0025	JMP	0020	; Jump to 0020 (repeat cycle)

AKN Acknowledge Single Input

```
E  0860 AKN
12  1
```

12	-	Auxiliary input number (01-07)
1	-	Input status
		0= input status **Off**
		1= input status **On**

With the AKN command, the DLC scans the status of the programmed auxiliary input for the specified state. Stepping to the next block will **not** take place until the desired status is present at the specified auxiliary input.

Note: Auxiliary input numbers 8-37 can also be used with optional DEA 5 & DEA 6 expansion I/O card.

Note: Input 00 does not exist. AKN commands containing Input 00 result in an “Invalid Program Command” diagnostic.

Example:

0000	JMP	0860	; Jump to block 0860
0860	AKN	02 1	; Scan aux input 02 until status is ON
0861	PSI	1 +00010.56 500	; axis 1 incremental feed, with acknowledgment, of +10.56 IU at 50% of maximum velocity (A100)
0862	AEA	11 1	; Turn On auxiliary output 11
0863	JMP	0900	; Unconditional Jump to block 0900
900	AKN	02 0	; Scan aux input 02 until status is OFF
901	AEA	11 0	; Turn OFF output 11
0902	PSI	1 -00010.56 999	; axis 1 incremental feed reverse, with acknowledgment, of 10.56 IU at 100% of maximum velocity (A100)
0903	JMP	0860	; Jump to block 0860 (repeat cycle)

AKP Acknowledge Parallel Input

```
E  0044 AKP
3     2100122011
```

3 - Bank number 0-9 (group of 10 inputs)

Bank X = Inputs X0-X9, i.e. Bank 2= Inputs 20-29, Bank 3= Inputs 30-39, etc.

2100122011 - input status (each of 10) as listed below:

0 = the input will be checked for condition **Off**

1 = the input will be checked for condition **On**

2 = the input will not be checked **- "Don't Care"**

The AKP command is used to verify the status of a specific bank, or group, of ten auxiliary inputs. Stepping to the next program block takes place after all inputs have met their programmed status simultaneously.

Note: Input 00, of Bank 0, does not exist. Program this input with a “Don’t Care” (2) status. No error diagnostic is issued if this input is programmed with anything besides a Don't Care condition.

Note: The standard DLC-A or DLC-G with the DEA 4 system/ auxiliary I/O card has a maximum of 7 auxiliary inputs that are physically accessible to the user. The DLC-A or DLC-G with the DEA 5 expansion I/O card has expanded auxiliary input capability of 22 inputs. The DLC-A or DLC-G with the DEA 5 & DEA 6 expansion I/O cards has expanded auxiliary input capability of 37 inputs.

The following is an example use of the AKP command:

The AKP command waits until all the specified aux. inputs, 10-19, have achieved the indicated status shown by the data field "0110201222".

```
E  0044 AKP
1     0110201222
```

Bank (tens) =	1	1	1	1	1	1	1	1	1	1
	|	|	|	|	|	|	|	|	|	|
Individual # (ones) =	0	1	2	3	4	5	6	7	8	9
	|	|	|	|	|	|	|	|	|	|
Input Number =	10	11	12	13	14	15	16	17	18	19
Programmed state	0	1	1	0	2	0	1	2	2	2

Inputs 10, 13 and 15 are checked for condition "OFF"

Inputs 11, 12, and 16 are checked for condition "ON"

Inputs 14, 17, 18 and 19 are not checked for a condition (Don't Care)

Example:

0000	NOP		
0001	AEA	01 1	; Turn ON output number 1
0002	AKP	1 0110201222	; Scan inputs 10-19 until programmed state is matched
0003	AEA	01 0	; Turn OFF output number 1
0004	JST	0001	; Jump to block 0001 and stop

If auxiliary inputs are not physically accessible with the current configuration of the auxiliary inputs and anything besides a Don't Care (2) condition is programmed, the program would hang-up because these inputs would not be satisfied.

APE Activate Parallel Outputs

```
E  0044 APE
1      2100122011
```

1 - Bank number 0-9 (group of 10 aux. outputs)

Bank X => aux. outputs X0-X9, i.e. **Bank 2**= aux. outputs 20-29, **Bank 3**= aux. outputs 30-39, etc.

2100122011 aux. output status (each of 10) as defined below:

0 = the aux. output will be reset to an **OFF** condition

1 = the aux. output will be set to an **ON** condition

2 = the aux. output will not be changed

The APE command sets the state of any programmed bank, or group, of ten aux. outputs. The desired bank of aux. outputs to be manipulated is first selected. The next ten digits set the status of each individual aux. output in the bank. Stepping to the next block takes place immediately after the APE command is read. The standard DLC has aux. output numbers 1-11. You can use aux. outputs number 12 through 88 as flags (bit memory).

Warning: The manipulation of aux. outputs 89 through 99 can have unexpected results. Refer to Table 5.1 for more information.

Note: Auxiliary Output 00, of Bank 0, does not exist. Program this aux. output with a “Don’t Change" (2) status. No error diagnostic occurs if this output is programmed with anything besides a Don't Change condition.

Note: The standard DLC-A or DLC-G with the DEA 4 system / auxiliary I/O card has a maximum of 11 auxiliary outputs that are physically accessible to the user. The DLC-A or DLC-G with the DEA 5 expansion I/O card has expanded auxiliary output capability of 27 outputs. The DLC-A or DLC-G with the DEA 5 & DEA 6 expansion I/O cards has expanded auxiliary output capability of 43 outputs.

The following is an example of the APE command:

The aux. outputs in Bank 1, aux. outputs 10-19, will be programmed to the states designated in the data field "2100122011" respectively.

```
E  0044 APE
1      2100122011
```

Bank =	1	1	1	1	1	1	1	1	1	1
	|	|	|	|	|	|	|	|	|	|
Individual # =	0	1	2	3	4	5	6	7	8	9
	|	|	|	|	|	|	|	|	|	|
Aux. Output Number =	10	11	12	13	14	15	16	17	18	19
Programmed State	2	1	0	0	1	2	2	0	1	1

Aux. Outputs 12, 13 and 17 are programmed for an "OFF" state.

Aux. Outputs 11, 14, 18, and 19 are programmed for an "ON" state.

Aux. Outputs 10, 15 and 16 are programmed with a Don't Change condition. This causes the output to remain in its previous state.

APJ Activate Parallel Output, then Jump

```
E  0304 APJ
1000
1 2100221011
```

1000 - Target block (0000-2999)

1 - Bank number 0-9 (group of 10 aux. outputs)

Bank X => aux. outputs X0-X9, i.e. Bank 2= aux. outputs 20-29, Bank 3= aux. outputs 30-39, etc.

2100122011 aux. output status (each of 10) as listed below:

0 = the output will be reset to an **Off** condition

1 = the output will be set to an **On** condition

2 = the output will not be changed

This command can be used to simultaneously set the condition 10 auxiliary outputs of the DLC.

The jump to the target block takes place after all 10 auxiliary outputs meet the programmed condition.

Notes:

1. Output 00, of Bank 0, does not exist. Program this aux. output with a “Don’t Change" (2) status. No error diagnostic occurs if this output is programmed with anything besides a Don't Change condition.
2. **The** standard DLC-A or DLC-G with the DEA 4 system/ auxiliary I/O card has a maximum of 11 auxiliary outputs that are physically accessible to the user. The DLC-A or DLC-G with the DEA 5 expansion I/O card has expanded auxiliary output capability of 27 outputs. The DLC-A or DLC-G with the DEA 5 & DEA 6 expansion I/O cards has expanded auxiliary output capability of 43 outputs.

Warnings:

1. If the Monitoring Window is turned OFF by setting flag 95 to ON, the DLC will have no way of detecting if a motor has Drive Runaway or Drive Stalled.
2. If **motion** is interrupted by setting flag 98 to ON, it will resume automatically when the output is turned OFF.

The following is an example of the APJ command:

The aux. outputs in Bank 2, outputs 20-29, will be programmed to the states designated in the data field "2100122011" respectively. The program then jumps to the desired block designated by the data "1000" in the example.

```
E  0044 APJ
1000
2 2100122011
```

1000 - After the designated bank of aux. outputs have been programmed, the program jumps to this block.

Bank 2 =	2	2	2	2	2	2	2	2	2	2
	|	|	|	|	|	|	|	|	|	|
Individual # =	0	1	2	3	4	5	6	7	8	9
	|	|	|	|	|	|	|	|	|	|
Aux. Output Number =	20	21	22	23	24	25	26	27	28	29
Programmed state	2	1	0	0	1	2	2	0	1	1

Outputs 22, 23 and 27 are programmed for an "OFF" state.

Outputs 21, 24, 28, and 29 are programmed for an "ON" state.

Outputs 20, 25 and 26 are programmed with a Don't Change condition. This causes the output not to change state.

The aux. outputs shown, 20-29, only serve as flags if a DLC-A or DLC-G with DEA 4 is being programmed. If the DLC-A or DLC-G with DEA 4, 5, and 6 is being programmed, the aux. outputs up to 43 are accessible.

ATS Acknowledge Output Status

```
E  0050 ATS
05   1
```

05	-	Auxiliary output number scanned (01-99)
1	-	Auxiliary output status
		0= output status Off
		1= output status On

The ATS command scans the status of the programmed auxiliary output. Stepping to the next block does not take place until the desired status is present at the specified auxiliary output. This command is used to perform handshaking with a parameter specified output or with an output whose state is changed by another task number.

All auxiliary outputs can be monitored and used in the execution of an ATS command. Auxiliary outputs 73 through 99 function differently than other aux. outputs. See Table 5.1 for more information.

The following example shows how the ATS command is used to test axis position. The program can be held up until the moving axis is in position by monitoring the "In Position" output programmed in Parameter A106 for each axis.

Example:

The "In Position" output programmed for the axis is number 15 (Aux. output 15 is Off while the axis is in motion until it is "In Position" at which time it will come On).

0000	JMP	0500	;Jump to Block 0500.
0500	POI	1 +00100.15 39.5	;Incremental position command, axis 1, +100.15 IU at 39.5% of max. feed rate.
0501	ATS	15 1	;Poll status of aux. 15 for an On status (axis has reached position).
0502	AEA	02 1	;Turn On aux. output 02.
0503	JST	0000	;Jump and Stop at Block 0000.

BAC Branch And Count

```
E  0060 BAC
1500 +0000 00500
```

1500 - Target Block (0000-2999)

+0000 - Offset Value

00500 - Preset Count

Each time this block is executed, the Actual Count increments by one. Program branching to the Target Block occurs until the Actual Count equals the Preset Count. Then, the Actual Count is set to zero and stepping to the next program block number takes place. See Counter Display screen example below.

Information about the Offset Value:

+1234 - Add Offset Number of items to Actual Number of items

-1234 - Subtract Offset Number of items from Actual Number of items

00000 - Reset Actual Number of items to zero

+0000 - Actual Number of items unchanged

-0000 - Actual Number of items unchanged

This is an example of the Counter Display screen (See Chapter 2 for procedures to scroll through displays and view the Counter Display screen).

```
         ______ Program block number of BAC command
    ____|
A: 1500 Counter
  000056  000500
     |         |
     |         |
     |         |
     |         |_ Preset Counter Value
     |
     |_ Actual Count
```

Immediately after this block is stored in memory, the Offset Value entered is summed with the Actual Count and becomes the New Actual Count. Then, the Offset Value in the BAC program block display is set to "+0000." This ensures that the Actual Number of items is only affected once (for instance, if Store is pressed again, the actual count will not be changed).

The jump to the Target Block continues to be executed until the programmed Preset Count has been reached. After that, the DLC steps to the next block and the Actual Counter is set to zero. The Actual Count can also be set to zero by using the program command "CLC".

Example:

Assume the Preset Count desired is normally 100. At this time, the Actual Count is 50. However, you desire only 20 more pieces, 70 total. Program the BAC block as follows:

The Offset Number data field should be programmed with a "+0030".

```
E  0600 BAC
1500 +0030 00100
```

When you press the Store key, the Actual Count is set to 80 (when viewing the Counter Display screen). Therefore, the Preset Count is reached after 20 counts, resulting in 70 total parts. All times after that, the counter starts at 000000, resulting in 100 increments.

The same holds true if a negative number is entered in the Offset Number data field of the BAC program block. Using the same example as above, if an Offset Number of <-0030> was entered, the Actual Count starts at 20. Therefore, 80 increments are encountered before the 100 Preset Count is achieved. Thereafter, the Actual Count begins at 000000.

If a negative Offset Number entered is greater than the Actual Count, the Actual Count will be set to zero. The Actual Count cannot be set to a negative value (set to do less than zero increments).

An example use for this command would be when you need to make up for bad parts. Let's say you had a jam, reached the end of the roll of material, or for some other reason you have five defective parts. Now instead of 100 parts, you need 105 parts to make your production quota. There are two things that can be done to the program block containing the BAC command. The Preset Count can be changed from 100 to 105 parts or <-0005> can be entered as an Offset Value to yield five extra parts (by decrementing the Actual Count by five parts). If the Preset Count is modified, a new Preset Count may need to be entered for proper quantity of parts per quota.

BCA Branch Conditional on Acknowledgment (Output-Dependent)

```
E  0700 BCA
0345  22  0
```

0345	-	Target block (0000-2999)
22	-	Auxiliary output number (01-99)
0	-	Jump condition
		0= jump if aux. output is **Off**
		1= jump if aux. output is **On**

A jump is executed if the programmed output meets the preselected condition (0/Off or 1/On). If the condition is not met at the programmed output, the program steps to the next block, instead of branching to the target block.

Example:

Assume that the following conditions exist: a part requires a constant absolute positioning by the axis to +8.125 inches. Then, the program needs to jump to one of two places, depending on the status of output 20, the axis will move to a certain position. In this example, output 20 is turned ON in another part of the program. To run this program, the axes must be homed.

0000	JMP	0053	; Jumps to block 0053
0053	PSA	1 +8.125 999	; axis 1, absolute positioning to +8.125 in. at max. feed rate.
0054	BCA	0100 20 1	; Jump to block 100, if output 20 is ON (ON in this example)
0055	BCA	0200 20 0	; Jump to block 200 if aux. output 20 is OFF
0056	JMP	0054	; Program execution could progress to here, if output 20 turns ON after block 0054 executes, scan output 20 again
0100	PSA	1 +5.25 75.5	; axis 1 absolute position to +5.25 in. at 75.5% max. feed rate
0101	JST	0053	; Jump to beginning of cycle at block 0053 and stop
0200	PSA	1 +0.000 500	; axis 1 absolute position to 0.0 IU at 50% velocity
0202	JST	0000	; Jump to beginning of cycle and stop

BCB Binary Conditional Branch (Inputs)

```
E  0800 BCB
1000  12 1
```

1000	-	Block Offset (0000-2999)
12	-	Length of jump
1	-	Auxiliary input bank selection (can be 1, 2, or 3)
		(This determines the location of the Binary inputs)

The BCB command executes a jump which has been defined by means of the Binary Inputs at the aux. inputs 1 to 8, relative to input bank selection.

The target block is calculated as follows:

Target Block = Offset + (Binary Input Value x Length of Jump)

The Binary Input Value should be converted to a decimal value to calculate the target block.

The Binary input location can be selected from the following:

Input Bank Selection 1 - use auxiliary inputs 1-4

Aux. Input:	4	3	2	1
Binary Significance:	2^3	2^2	2^1	2^0
Decimal Significance:	8	4	2	1

Input Bank Selection 2 - use auxiliary inputs 5-8

Aux. Input:	8	7	6	5
Binary Significance:	2^3	2^2	2^1	2^0
Decimal Significance:	8	4	2	1

With input bank selection 1 or 2, a total of 15 targets is possible.

Input Bank Selection 3 - use auxiliary inputs 1-8

Aux Input:	8	7	6	5	4	3	2	1
Binary Significance:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Significance:	128	64	32	16	8	4	2	1

With input bank selection 3, a total of 256 targets is possible.

Note: There is no "Invalid Program Command" diagnostic if this command is programmed to exceed block 2999. That is because the destination block is dependent on the value determined from aux. inputs (relative to the Binary Input Value selected). However, an "Invalid Block #" diagnostic is issued when the BCB command is executed and the destination block exceeds program block 2999.

BCD Binary Coded Decimal Conditional Branch (Inputs)

```
E  0900 BCD
0100  12
```

0100 - Block Offset (0000-2999)

12 - Length of jump

The BCD command executes a jump which has been defined by means of BCD (Binary Coded Decimal) inputs at the auxiliary inputs 1 to 8.

The target block is calculated as follows:

Target Block = Offset + (BCD Input Value x Length of Jump)

The BCD Input Value should be converted to a decimal value to calculate the target block.

Example:

If a parallel input is not sensed in the BCD format, this will result in "BCD Input Error" diagnostic.

Aux. Inputs:	8	7	6	5	4	3	2	1
BCD Significance:	80	40	20	10	8	4	2	1
Aux. Inputs:	8	7	6	5	4	3	2	1
BCD Input:	0	1	0	1	0	0	1	1
Decimal Equivalent:	50			+		3		

Target Block: 0100 + (53 x 12) = 736

In the example above, a jump to block 736 takes place.

Aux. Inputs:	8	7	6	5	4	3	2	1
BCD Input:	0	0	0	1	0	0	0	1
Decimal Equivalent:	10			+		1		

Target Block: 0100 + (11 x 12) = 232

If BCD input(decimal equivalent) = 00 - jump to block 0100

01 - jump to block 0112; offset of 100 + (1 x 12)

66 - jump to block 0892; offset of 100 + (66 x 12)

Note: There is no "Invalid Program Command" diagnostic if this command is programmed to exceed block 2999. That is because the destination block is dependent on the value determined from aux. inputs (relative to the BCD value selected). However, an "Invalid Block #" diagnostic is issued when the BCD command is executed and the destination block exceeds program block 2999.

BCE Branch Conditional on Single Input

```
E  0100 BCE
0234  01   0
```

0234 - Target block (0000-2999)

01 - Auxiliary input number (01-07)

Note: Auxiliary input numbers 08-37 can also be used with optional DEA 5 & DEA 6 expansion I/O card.

0 - Condition:

0 = jump if input is OFF/low

1 = jump if input is ON/high

A conditional jump to the target block will be executed if the auxiliary input meets the preselected condition (0 or 1). If the condition is not met at the programmed input, stepping to the next block takes place.

One use of this command is to allow selecting different routine programs to run for making different parts, by turning ON a different switch on the control panel. In the following example, the main program runs, then branches and executes the routine program indicated by the first input that is high (ON). The last block of the routine can jump back to the first block of the main program for continuous operation. It can jump back to the beginning of the main program (block 0000), and branch to the respective block for the input that is high.

Example:

0000	NOP		
0001	BCE	0600 01 1	; Branch to block 0600 if input 01 is high
0002	BCE	0700 02 1	; Branch to block 0700 if input 02 is high
0003	BCE	0800 03 1	; Branch to block 0800 if input 03 is high
0004	BCE	0900 04 1	; Branch to block 0900 if input 04 is high
0005	JMP	0000	; Jump to beginning of cycle at block 0000 (scan inputs 1-4 again)

BIC Branch Conditional On Binary Inputs

```
E  0801 BIC
0300 20 15 3 0
```

0300	Block Offset (0000-2999)
20	Length Of Jump
15	Starting Auxiliary Input Number (01 - 37)
3	Number of (Consecutive) Auxiliary Inputs Required (1-8)
0	Mode
	0 = Binary Mode
	1-9 = Future Options

The BIC command executes a jump which has been defined by the binary state of the selected number of inputs required multiplied by the length of jump, added to the target block. Binary mode is the only conversion mode allowed with the BIC command. The user selects the starting auxiliary input number and the number of consecutive binary inputs required.

The target block is calculated as follows:

Target Block = Block Offset + (Binary Input Value x Length Of Jump)

Example:

```
E  0000 BIC
0300 10 05 4 0
```

Aux. Inputs:	8	7	6	5
Binary Input:	1	0	0	1
Decimal Significance:	8	4	2	1

Decimal Equivalent = 9

Target Block = 0300 + (9 x 10) = 390

BIO Branch Input/Output Compare

```
E  0801 BIO
0321
2 1111222222
```

0321	-	Target Block (0000-2999)
2	-	Bank number 0-9 (group of 10 aux. inputs/outputs) Bank X => aux. I/O X0-X9, i.e. Bank 2= aux. I/O 20-29, Bank 3= aux. I/O 30-39, etc.
1111222222		aux. inputs/outputs states for comparison. States are defined below: 0 = I/O compared for OFF condition 1 = I/O compared for ON condition 2 = I/O not compared - Don't Care

This command is used to check the aux. inputs identified with "1" for the level input "1" if the aux. outputs of the same number have been set to "1". The jump to the target block takes place when the condition is met.

The inputs in the BIO data marked with "0" or "2" will not be checked. The only comparison that will cause a jump is if each respective BIO data = "1", output = "1" and input = "1".

Example:

```
E  0801 BIO
0400
1 1110000000
```

Inputs and outputs from 10 to 13 will be compared

BIO Data	10="1"	11="1"	12="1"	13="0"	14="0"
Outputs	10="1"	11="1"	12="1"	13="0"	14="1"
Inputs	10="1"	11="1"	12="0"	13="1"	14="1"
Condition	met	met	not met	not checked	not checked

Since the condition is not being met at one point (input 12), the jump will not be executed. Stepping to the next program block takes place instead.

BMB Branch on Multiple Binary Outputs

```
E  0120 BMB
1234 45 67 8
```

1234	-	Block Offset (0000-2999), used to calculate Target Block
45	-	Length of Jump
67	-	Starting Output Number (01-99)
8	-	Number of Outputs (consecutive) Being Used

This command will cause a jump to be executed which is determined by means of the output assignment as defined in the command.

Example:

```
E  1200  BMB
0100 02 05 8
```

Outputs:	12	11	10	09	08	07	06	05
Significance:	128	64	32	16	08	04	02	01
Output value:	0	0	1	1	0	0	1	1

Output value = 51

The target block is calculated as follows:

Block offset + (output value x length of jump)

0100 + (51 x 02)

Target block = 0202

BPA Branch on Parallel Acknowledgments (Outputs)

```
E  0130 BPA
0234
1 1111000022
```

0234	-	Target Block Number (0000-2999)
1	-	Bank number 0-9 (group of 10 aux. outputs)
1111000022	-	Output State (each of 10) as listed below:

0= the output will be checked for condition Off

1= the output will be checked for condition On

2= the output will not be checked - Don't Care

This command represents an expansion of the command BCA. It can be used to check if a condition is being met at 10 auxiliary outputs. The condition can be specified separately for each output.

The jump to the target block takes place only if all 10 aux. outputs meet their programmed condition. If not, stepping to the next block takes place.

BPE Branch on Parallel Inputs

```
E  1500 BPE
0345
6 1111000022
```

0345 - Target Block

3 - Bank number 0-3 (group of 10 aux. inputs)

1111000022 - Input State (each of 10) as listed below:

0= the input will be checked for condition OFF

1= the input will be checked for condition ON

2= the input will not be checked - Don't Care

This command represents an expansion of the command "BCE". It can be used to check simultaneously if a condition is being met at 10 auxiliary inputs. The condition can be specified separately for each input.

The jump to the target block takes place only if all 10 inputs meet the programmed condition. If not, stepping to the next block takes place.

BPT Branch on Position Test

```
E  0060 BPT
0456
1 +12345.678
```

0456	-	Target Block
1	-	Axis 1
+12345.678	-	Absolute Position (2 or 3 decimal places, as set in B007)

This command can be used to check the absolute position of the axis. If the axis is in position (+/- switching tolerance set in A106), the jump to the target block will take place. If not, stepping to the next block takes place.

Note: The BPT command only functions after the axis has been homed. Prior to this, the block is scanned only and no execution takes place.

BZP Branch If the Target Position Exceeds the Position Limit Value

```
E  1550 BZP
0678
1 +12345.678
```

0678	-	Target Block
1	-	Axis 1
+12345.678	-	Position Limit Value (to 2 or 3 decimal places, as set in B007)

You can use this command to check that a commanded positions does not exceed the position limit value set by this command.

At the execution of this block, the jump to the target block takes place if the actual position, or the target position, of the axis, is equal to or greater than, the programmed position limit value.

If the actual position, or the target position, is less than the programmed position limit value, the program steps to the next block, instead of jumping to the branch block number.

Note: The BZP command only functions after the axis has been homed. Prior to this, the block is scanned only and no execution takes place.

CID Change Instruction Data

```
E  0034 CID
1234
1 0 +0000001
```

1234 - Target block in which the information is to be changed (0000-2999)

1 - Info position:

1= The first modifiable variable value in the target block is processed (e.g., the length in POI command)

2= The second modifiable variable value in the target block is processed (e.g., the velocity in POI command)

0 - Operation value added or subtracted

0= Add or subtract operation value (depending on sign of change value)

1= Same as 0, but the result of the operation must be positive, otherwise the operation will not be carried out

2= Same as 0, but the result of the operation must be negative, otherwise the operation will not be carried out.

3= Overwrite information with the prefix in the target block

+ - '+'= the change value is added to target value

'-' = the change value is subtracted from the target value

0000001 - The information in the target block (specified by the entry for 'type of information') is changed by this amount. The position of the decimal is not considered!

This command changes the data in another block. Stepping to the next block takes place when the data changes in the target block have been completed or when, due to restrictions, no change can be carried out. If the target block includes no valid command, stepping to the next block will take place without any changes in the target block. For CON and FOL, designation of the information type has no significance.

NOTES:

1. It is only possible to process one CID, KDI or WRI command at a given time. The commands waiting in other tasks will be delayed until completion of the first instruction.
2. At present, the use of the CID command is limited to the CON, FOL, POI, PSI, POA and PSA commands. All others commands will be bypassed with no change.

Command	Info Position	
	1	2
CON speed in %	Speed in %	
FOL Follow factor	Follow factor	
POI, PSI, POA, PSA	Length	Speed in %
PST, JMP, VCA, WAI		

The information is transmitted to the target statement in right-justified format. In each case, the following command is called once: CID 1234 1 0 +0000212.

CIO Copy Input/Output to Output

```
E  0160 CIO
1  03  75  9
```

1 - 0 = Copy source Auxiliary Inputs
    1 = Copy source Auxiliary Outputs
    2 = Copy source System Inputs
    3 = Copy source System Outputs

03 - First Input/Output Number looked at (Copy Source, 01 to 99)

75 - First Output Number to Copy to (Copy Target / Destination, 01 to 99)

9 - Quantity of I/O to be Copied (1-9)

This command can be used to copy the status of several inputs or outputs to other outputs.

Warning: The manipulation of aux. outputs 89 through 99 can have unexpected results. Refer to Table 5.1 for more information.

Example:

Five inputs are copied to five outputs;

```
E  0160 CIO
0  01  23  5
```

Results of this command can be monitored using the input/output status display.

Status of the inputs are displayed as:

```
DLC Inputs 1-7
1.1.1..
```

----- Status of inputs 1-5 is 1.1.1 = 10101

Status of the outputs before the command is executed:

```
DLC Outputs12-27
1..11..111.111..
```

----- Status of outputs 23-27 is 01110

Status of the outputs after the command is executed:

```
DLC Outputs12-27
1..11.1.1..1.1.1
```

----- Status of outputs 23-27 is now 10101

Rexroth Indramat

CLA Clear Absolute Position

```
E  1700 CLA
1
```

1 - Axis 1

The record of absolute position is cleared for the selected axis using this command.

Multi-turn absolute encoders:

- Parameter A110=xxx2xxxx
  The instantaneous position is reset to zero with the “CLA” command. An off set dimension from parameter A111 is **not** considered. This results in the current evaluated position becoming=zero.

Incremental encoder systems:

- Parameter A110=xxx1xxxx
  The instantaneous position is reset to zero using the “CLA” command. An off set dimension from parameter A111 **is considered**. This results in the current evaluated position thereby becoming=offset from parameter A11.
- If parameter A110=xxx0xxxx, the fault report “Illegal Command” is generated when this command is called.
- The default delay time for indexing to the next statement is 1 millisecond.

CLC Clear Counter

```
E  0037 CLC
0123
```

0123 - Counter block number (0000-2999)

Use this command to clear (set to zero) the actual value of the counter at the indicated block number. If the indicated block does not contain the count command "COU" or "BAC", this block is only scanned.

The default delay time for indexing to the next statement is 1 millisecond.

COC Cam Output Control

```
E  0020 COC
1 05 110022 +090
```

1 - Axis 1

05 - Number of the first of six aux. outputs to be set

110022 - Status of the 6 aux. outputs starting with the first:

0= resets output to Off

1= sets output to On

2= output remains unchanged

+090 - Test position of axis in Input Units to zero or one decimal place (can be + or -)

- If B007 is set for 2 decimal places (B007= xx02xxxx), the test position has no decimals (i.e. 360= 360).
- If B007 is set for 3 decimal places (B007= xx03xxxx), the test position has one decimal (i.e. 360= 36.0).

The COC command sets a bank of six aux. outputs relative to the axis position in degrees. The DLC waits for the programmed axis to reach the test position before setting the six aux. outputs to the desired states. The first of six consecutive aux. outputs to be changed is entered in the program block. The DLC then advances to the next program block.

Due to the scan time, the delay of setting the output states compared to the axis position takes 1 millisecond. Therefore, the actual position of the axis when the outputs are set can vary with the present velocity of the axis. The formula used to calculate the accuracy of the outputs is:

$$\frac{\text{Speed in EGE/sec.}}{1000} \quad \text{x} \quad 1\ \text{msec}$$

If the axis has been homed, all test positions are taken in reference to the current absolute position. If the axis has not been homed, the test position is taken in reference to the last incremental feed.

Set up A116 for Rotary Table, A108 for 360 degree feed constant and the system implements an incremental encoder for position feedback.

0000	CLA	1	;Sets actual position counter for axis 1 to zero (initializes axis).
0001	APE	0 2000000222	;Sets all cam outputs (aux. outputs) 1 through 6 to the Off state.
0002	CON	1 1 +001 00	;Sets axis 1 into continuous motion at 0.1% of max. velocity.
0003	COC	1 00 221122 +365	;Waits until the 5-degree or 365 degree position has been reached and then turns On cam outputs (aux. outputs) 3 and 4.
0004	COC	1 01 022222 +040	;Waits until the 40-degree position has been reached, and then turns output 1 Off.
0005	COC	1 01 220222 +150	;Waits until the 150-degree position has been reached, and then turns output 3 Off
0006	COC	1 01 222022 +180	;Waits until the 180-degree position has been reached, and then turns output 4 Off.
0007	COC	1 01 212222 +305	;Waits until the 305-degree position has been reached, and then turns output 2 On.
0008	COC	1 01 202222 +335	;Waits until the 335-degree position has been reached, and then turns output 2 Off.
0009	COC	1 01 122222 +352	;Waits until the 352-degree position has been reached, and then turns output 1 On.
0010	JMP	0003	;Jump to block 0003.

CON Continuous Operation

```
E  1900 CON
1  1 +345 01
```

1	-	Axis 1
1	-	Start/stop continuous operation (0=stop, 1=start)
+345	-	Speed in % to 1 decimal place
		+/- direction (relative to Direction of Operation Parameter A109)
01	-	Output (auxiliary function) to turn ON when CON velocity is achieved
		00 = no output

Warning: Subsequent commands relating to the axis programmed here may be affected by the continuous operation. If the axis has limited travel (i.e. ballscrew), safeguards should be taken to assure the axis will be stopped before travel limit is exceeded.

An output can be programmed which is set to "1" once the continuous operating velocity has been reached. The axis position commands should not be used while continuous operation is on. This command will not be executed for an axis with an absolute value encoder.

The following example could be used on a web application to run speeds to keep a loop storage accumulator filled:

0000	JMP	0100	; Jumps to block 0100
0100	CON	1 1 +750 00	; Feed at initial Fill Speed (75%)
0101	BCE	0110 02 1	; Branch to 0110 if first loop light (input 02) is covered/input high (normal running),
0102	BCE	0120 03 1	; Branch to 0120 if second loop light (input 03) is covered/input high (loop full)
0103	JMP	0101	; Scan inputs 02 and 03 again (continue fill speed)
0110	CON	1 1 +500 02	; Change speed to 50%, turn ON "loop normal" output
0111	BCE	0100 02 0	; Branch if accumulator storage is low
0112	BCE	0120 03 1	; Branch if accumulator is full
0113	JMP	0111	; Scan again for low/full conditions (inputs 02 and 03)
0120	CON	1 0 +000 00	; Stop feeding, accumulator is full
0121	AEA	03 1	; Turn ON "storage full" output
0122	AKN	03 0	; Wait for additional material to be removed to uncover loop full sensor
0123	AEA	03 0	; Turn OFF "storage full" output
0124	JMP	0110	; Resume at normal running speed (block 0110)

COU Count

```
E  2000 COU
+12345 12 123456
```

+12345 - Offset number of items

12 - Signal (aux.) output number (01-88)

123456 - Preset number of counts

Once this block has been read, the actual item number is increased by 1. Once the preset number of counts has been reached, the programmed output is switched on. The actual value is subsequently cleared. The item counter can be also set to zero by using the command 'CLC.' Any number of item counters can be programmed.

For an example of how to offset the actual count, refer to the example for the BAC command.

This is an example of the Counter Display screen (See section 2.3.5 for information how to view the Counter Display screen).

```
       _______ Program Block Number of COU command
  ____|
A: 2000 Counter
  000056  000500
     |       |
     |       |
     |       |
     |       |____ Preset Counter Value
     |
     |_______ Actual Count
```


Rexroth Indramat

CPL Clear Position Lag

This command is used to clear the following error, after a move to a positive stop routine has been completed.

```
E  0201 CST
1  4
```

The lag error of the selected axis is zeroed once.

The program moves the next block immediately following this command.

CAUTION: This command directly affects the position control loop. Any lag error that is build up will be deleted.

CST Change Subroutine Stack

```
E   0201 CST
1   4
```

1 - Changes subroutine level (0-2)

0= changes subroutine level in Task 1 and Task 2.

1= changes subroutine level in Task 1 only.

2= changes subroutine level in Task 2 only.

4 - Set subroutine stack to this level

0= sets stack to zero (See Note below.)

1-9= change subroutine stack level from one to nine levels

The starting block for Task 2 is set up in "Parameter B006 - Start Task 2 & 3". Task 1 programming does not need to be set up in parameters.

The CST command is used to change to different subroutine stack levels. It is used in conjunction with JSR and RTS. Task 1 and Task 2 have a maximum number of nine stack levels that can be changed in a single CST command. The CST command can be used in subroutines to allow for more functions when program space is limited.

After using a CST command to correct a stack, the next RTS command will jump the program beyond undesirable subroutines to a selected level.

Note: If the stack is set to zero, an RTS command cannot be used as this becomes the programs main level. If an RTS command is used after setting the stack to zero, the error "RTS Nesting" will occur.

Example:

The following example shows the program flow and the CST command (Subroutine level 3, block, "E0309 CST 1 1", is used to change the subroutine stack level to level 1, in Task 1).

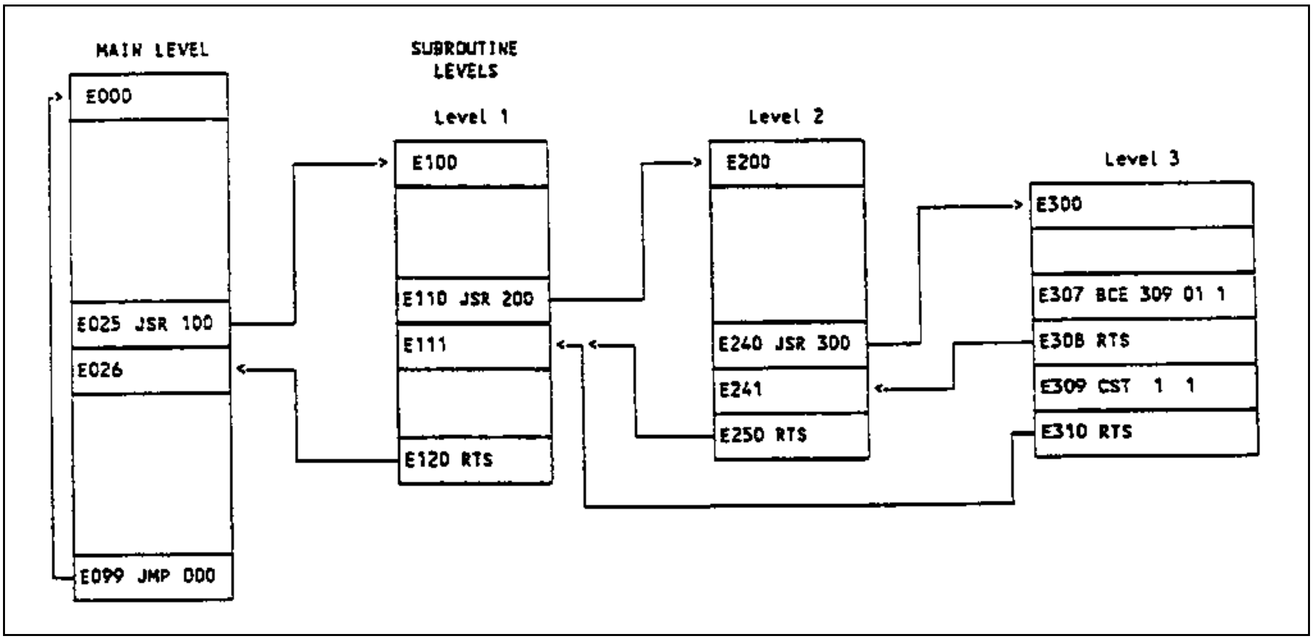


Figure 5-1: Program Flow, CST Command

Rexroth Indramat

D== Tool Correction Memory Selection

```
E  1000 D==
nn
```

nn - Number of the correction memory selected (00...19)

An activated tool correction is effective for **all** feed commands (G01, POA, POI, POM, PSA, PSI, and PSM).

Registers D01-D19 can contain values up to seven digits. The eight digit is designated as the sign digit, either + or -.

Register D00 deactivates Tool Correction and unselects any active register. After a D==00 block, in order to activate Tool Correction, you must re-select a Tool Correction register using the D== command.

See Entering Tool Corrections Section at the end of Chapter 5.

FAK Factor All Motions

```
E  0210 FAK
1  1.999999
```

Global position correction (factor all feeds)

1 - Axis 1

1.999999 - Multiplication Factor - 0.000000 to 1.999999

This programmable factor is used to multiply the feed length or position in feed commands. The feed rate is not changed.

At the start of Automatic operating mode, the factor = 1.0000 until it is changed by the command FAK. The factor remains in effect until a new value is read with the command FAK during program execution.

Example:

```
E  000  PSI
1 +00010.000 200
```

= Position Axis 1 = 10.0 inches at 20% feedrate

```
E  001  WAI
01.00
```

= Wait 1.00 seconds

```
E  002  FAK
1  0.500000
```

= Multiply position by 0.5

```
E  003  PSI
1 +00010.000 200
```

= Position Axis 1 = 5.0 inches (0.5 x 10.0) at 20% feedrate

```
E  004  JST
000
```

= Jump to block 0000 and stop program.

A new value for the factor has no effect on the positioning currently being executed. The FAK factor takes affect on the next position command.

FUN Functions

```
E  1234 FUN
0 0 1 1000
```

0 - Unassigned, set to 0

0 - Unassigned, set to 0

1 - Feed length measurement, measuring wheel (0 = off, 1 = on)

1 - Feed length measurement, axis 1 (0 = off, 1 = on)

0 - Unassigned, set to 0

0 - Unassigned, set to 0

0 - Unassigned, set to 0

Significance of the inputs:

0 = Temporarily store measured value and then clear

1 = Clear the measurement value (start new measurement). A temporarily stored value is retained.

2 = No change

Status 46 is used to request the temporarily stored measurements via the serial interface.

Example:

The feed length traversed by the axis is to be determined after 10 events.

```
0000    NOP
0001    AKN  01 1
0002    FUN  0 0 0 1000
0003    AKN  01 0
0004    AKN  01 1
0005    BAC 0007 +0000 0009
0006    JMP 0009
0007    BCE 0005 01 0
0008    JMP 0007
0009    AKN   01 1
0010    FUN  0 0 0 0000
```

F== Feedrate

```
E  1000 F==
010000.00
```

010000.00 - Feedrate in units/min (to 2 or 3 decimal places, depending on setting in B007)

Range of values: 0...F_{max} (F_{max} = Parameter A100 x 60)

If F_{max} is exceeded, this command leads to the error message ***"F== Command"***

The "F== " instruction is valid for all feed commands (including PSA, POM, PSI...).
When an "F== " feedrate is programmed, the % of Vmax feedrate active in the program blocks up to that point is then ignored. This "F== " feedrate remains effective in memory until it is overwritten by a new "F== " value, or until it is canceled by the command: "F== 000000.00".

After processing a command having the contents "F==000000.00", the previous feedrate (in % of Vmax) is reactivated and becomes valid for all subsequent programming blocks. Once this occurs, it is no longer possible to use the G01 command.

G01 Linear Feed

```
E  1000 G01
1 ±000010.00 mm
```

1 - Axis (always 1!)

±000010.00 - Process path; absolute or incremental.

mm - Number of the M function (00...63) that is to be called up after reaching the target position. If no M function is used, 99 <u>must</u> be entered as the M function to be output. If 99>mm>63, the error message ***"ERROR M== Command"*** will be displayed.

This command causes a linear position feed at the previously specified feedrate. If the feedrate has not been previously defined with the "F== " command, when G01 is called up, the error message ***"G01 without "F== "*** results.

There is a parameter table available for the definition of the 64 pre-defined M function. Output of one of the M functions corresponding to this table can be programmed in the command G01. The M function programmed in a G01 block is set after the programmed position has been reached (within the limits set by A106, Position Tolerance). If G01 is programmed within a G61 contouring mode, the M function is set so as to allow optimum speed. Please see Section 4.6 of Chapter 4 for further information on M-Functions and their use.

With the G01 command, program block execution continues immediately after achieving the programmed position (within the position window A106).

G04 Dwell Time

```
E  1000 G04
tt.tt
```

tt.tt - Wait time in seconds

This command causes a delay in further execution of the program. It corresponds to the WAI command.

Program block execution is delayed by the programmed dwell time.

G40 Tool Correction Deactivated

```
E  1000 G40
```

This command disables Tool Correction within the process path in the programmed feed command.

G40 is effective upon entering automatic operation.

The register selected, remains selected, even after deactivating the Tool Correction with G40. If a G43 or G44 command is issued after a G40 is issued, the last selected Tool Correction register prior to the G40 command will be used as the selected value.

G43 Tool Correction - Added

```
E  1000 G43
```

Adds the contents of the active Tool Correction register to the programmed destination. The sign of the value in the Tool Correction register is considered in this computation.

Example:

If the value in the register is +1, this command will add +1 to the programmed destination. If the value is -1, this command will add -1 to the programmed destination.

G44 Tool Correction - Subtracted

```
E  1000 G44
```

Subtracts the contents of the active Tool Correction register to the programmed destination. The sign of the value in the Tool Correction register is considered in this computation.

Example:

If the value in the register is +1, this command will subtract +1 to the programmed destination. If the value is -1, this command will subtract -1 to the programmed destination (which means it will add 1 to the programmed destination).

G60 Exact Stop

```
E  1000 G60
```

If this command is invoked prior to a G01 command, all subsequent G01 commands are processed with exact stop (i.e., with Lag finishing). This means the execution of the next program block does not begin until the programmed position has been reached (within the position window parameter A106).

G60 is the default mode and is automatically active after restarting the program.

G61 Velocity Rate Optimization

```
E  1000 G61
```

This instruction enables processing of a velocity rate optimization profile using G01 and F== instructions. This profile may be up to a maximum of 20 blocks long and begins with the G61 command and ends with the G60 command.

This optimization prevents the axis from stopping between G01 positioning commands (without Lag finishing), even when the feed rate or program distance is changed. Within this profile, there may not be any programming commands that would prevent program execution from continuing immediately with the next block, (e.g., dwell time, program jumps, branch jumps, etc.). No change in direction of motion is possible within the profile. All positioning commands must be programmed only in absolute or only in incremental positioning. Any active Tool Correction is observed.

If a G60 programming command is encountered, the Velocity Optimization ends. If there is no G60 instruction after 20 blocks from the first G01 command, the error message ***"No G60"*** is displayed.

Program Example:

N0000 G91			;Set incremental positioning
N0001 G61			;Velocity rate optimization turned on
N0002 F==	010000.00		;Set F = 10 m/min
N0003 G01	1 +000200.00	01	;Move 200 mm and set M01
N0004 F==	005000.00		;Change F = 5 m/min
N0005 G01	1 +000100.00	99	;Move 100 mm. No M Function.
N0006 F==	002000.00		;Change F = 2 m/min.
N0007 AEA	01 1		;Set output 01 high
N0008 G01	1 +000050.00	02	;Move 50 mm and set M02.
N0009 G60			;Velocity rate optimization turned off.
N0010 G04	01.00		;1 second dwell time
N0011 JMP	0000		;Repeat cycle

G74 Reference Axis

```
E  1000 G74
a
```

a - Axis (always 1)

This command corresponds to the HOM command. All criteria and limits that apply to the HOM command also apply to the G74 command.

For further information, see the description for the HOM command.

G90 Absolute Dimensions

```
E  1000 G90
```

This command causes all subsequent process commands having a G code to be interpreted as absolute process commands.

G90 is the default and is effective upon entering automatic operation and after a program restart.

G91 Incremental Dimensions

```
E  1000 G91
```

This command causes all subsequent process commands having a G code to be interpreted as incremental process commands.

Once selected, G91 is effective for all subsequent blocks until it is unselected by a G90 command or until the program is restarted.

HOM Home Axis 1

```
E  0230  HOM
1
```

1 - Axis 1

Homing proceeds according to how parameters A110 - A114 have been programmed. Stepping to the next block takes place immediately after the block has been read in. Programming measures must be taken (ATS, AKN) to prevent absolute positioning commands (POA, PSA, etc.) from being encountered, before the homing sequence is completed, or an error message will be issued. An ATS command programmed to monitor the "Home Established" output (Parameter A112) will prevent stepping to the next block until the homing sequence is finished.

If the HOM command is used for the axis with an absolute value encoder, it will result in an error message.

Note: There are alternatives to Homing an axis, other than using this predefined routine. They are described in Appendix A, Programming Notes.

JMP Jump Unconditional

```
E  2500 JMP
1234
```

1234 - Target block

When this command is encountered in a given task, the program will immediately jump to the specified target block.

There are several other commands that also provide an unconditional jump, but provide added features in the same block. They can be useful when program space is limited. Refer to section 5.8, Command Summary, for alternatives.

JSR Jump to Subroutine

```
E  0250 JSR
1567
```

1567 - Start block of the subroutine

Subroutines can be called up at any time. The maximum permissible number of nested subroutines is 127.

A subroutine must end with the command "RTS" (program jumps back to the block following the JSR command). If an RTS command is not programmed at the end of a subroutine, the programmed lines following the end of the subroutine will be executed. This can result in unexpected axis movement, "Invalid Program Command" diagnostic, etc.

Note: Additional program flow control is possible by using the Change Subroutine Stack command (CST) with the JSR/RTS commands for advanced programming.

JST Jump and Stop

```
E  2600 JST
1678
```

1678 - Target Block Number

This command causes the program to jump directly to the target block and stop in that block. The DLC will issue a software $\overline{\text{Cycle Stop}}$ and stop the program, waiting in the target block for a new start impulse at system input "Cycle Start". The start impulse will cause the DLC to continue processing the program beginning with the target block.

A JST command executed in Task 1 will cause the program running in Task 2 to halt in its current block until a "Cycle Start" command is issued to restart the program.

JTK Jump in Task

```
E  1345 JTK
1000  1
```

1000 - Target block

1 - 1= Task 1
2= Task 2

The JTK command is used for initiating an Unconditional Jump within a selected task. This command terminates the current program sequence for the selected task and then causes it to jump to the target block. If the task is currently executing a move profile, the motion is completed before the task can be interrupted. If necessary, a PBK command can be used before the JTK to terminate the motion thereby providing an immediate interrupt. The JTK command permits the use of many specific interrupt routines trapped by numerous events.

Note: If only one (input dependent) interrupt routine is needed, it is recommended to use the interrupt vector parameter B012.

In most applications, the JTK command will be issued from one of the other tasks, typically Task 3. If used correctly, the JTK command can provide many of the functions available in B012 (Interrupt Vector) with the additional flexibility to use any event or series of events: position, count, one or more inputs or outputs to cause an interrupt to Task 1 or 2.

KDI Copy Position Difference

```
E  0500 KDI
2000  1000  1
```

2000 - Target block for the stored position difference

1000 - Compare position block number

1 - 0= Actual position minus the compared position
1= Compared position minus the actual position

The Compared Position is the difference between the Actual Absolute Position and an Absolute Position at a specific program block (i.e. block 1000, as shown above).

The KDI command will copy the position difference between the actual absolute position and the absolute position at a specific block number (Compare position block number). The position difference is then stored in the target block. The amount of time to process the KDI command can be up to 100 ms. The program will advance to the next block after the position difference is stored in the target block. The actual position is either subtracted from the compared position or the compared position is subtracted from the actual position.

In order for the KDI command to function properly, the following conditions must be met:

1. Axis 1 must be homed. If the axis is not homed when the KDI command is encountered in the program, then the error message "Axis 1 Not Homed" is displayed.

2. The compare position block number must contain an absolute position command (POA; PSA). If the program block is not an absolute position command, then the error message "Invalid Prog Cmd" will occur.

3. The target block must contain an incremental position command (POI; PSI). If the target block is not an incremental position command, then the position difference is lost permanently.

4. The WRI command cannot be processed while using the KDI command.

5. The KDI command can only be used in one task at a time.

MOM Torque Reduction

```
E  1345 MOM
1 020 040 01 123
```

1 - Axis 1

020 - Percentage Of Maximum Torque Until Positive Stop Is Reached
Minimum Percentage = 000
Maximum Percentage = 400

040 - Percentage Of Maximum Torque At Positive Stop
Minimum Percentage = 000
Maximum Percentage = 100

01 - Aux. Input for Torque Reduction
00 = No input needed, reduce torque on execution of MOM command.
01-07 = Reduce torque any time input is high after one execution of MOM command.

123 - Percentage Of Maximum Torque
Minimum Percentage = 000
Maximum Percentage = 400

Note: If the percentage of maximum torque values, entered in the MOM command, are greater than the maximum percentage allowed, the value entered will be interpreted as 100 percent.

The MOM command is used to limit the torque to the motor, when a positive stop is reached, or when MOM command is executed. The MOM command requires a PFI (Incremental Position To Positive Stop) or PFA (Absolute Position To Positive Stop) command to assist in positioning to a positive stop. The MOM command will determine the maximum torque allowed while positioning to the positive stop and the maximum torque allowed when the positive stop is reached. When the positive stop is reached, the programmed torque is applied until another position command (POI, POA, .etc) is executed.

Stepping to the next block takes place immediately after the MOM command is read in the program.

The percentage of maximum torques are valid for all PFA or PFI commands until a new MOM command redefines the percentages. The MOM command can be executed from another task.

When executing the MOM command from another task, the new percentages are valid immediately, even during the execution the PFA or PFI command in a separate task.

A typical application consists of programming the MOM command for the percentages of torque required and then commanding the axis to position, by using the PFI or PFA command. The commanded position must be greater than or equal to the positive stop position. The axis will travel to the programmed position, at percentage of maximum torque until positive stop is reached. The program flow will remain on the PFA or PFI program block until the positive stop is reached or the until the programmed distance, in the PFA or PFI command is completed.

If the programmed distance in the PFA or PFI command is completed before the positive stop is reached, the next program block will be executed. The next program block can be a jump (JMP) command to an error routine to tell the machine operator that the positive stop was not reached.

When the positive stop is reached, the percentage of maximum torque at positive stop will be enabled and the program flow will jump over the block immediately after the PFA or PFI command and the program will continue.

It is also possible to overwrite the torque limit while driving.

The default delay time for indexing to the next statement is 1 millisecond.

When using the MOM command, drive errors can occur at low torque limiting (“Excessive Position Lag”, “Drive Runaway”, drive error 78. “(“Excessive Position Lag” and “Drive Runaway” can be switched OFF in parameter A122. Rotational speed controller monitoring can be switched OFF using the PC program “DDS2PC.EXE” to prevent drive error 78.

Requirements For Recognizing a Positive Stop Condition:

1) Twenty milliseconds deceleration after reading the PFI command.
2) The positive stop is recognized when:
   a) Four times the following error (position lag) is reached.
   b) Axis movement is less than 1/1024 of a motor revolution per millisecond.

Note: If the positive stop is not reached, the maximum torque of the axis and the overload factor of the drive are active.

Warning: When the MOM command is used and while positioning to a positive stop or positioned at the positive stop, the overload factor in the drive is internally set to 400 percent. The MOM command is the only limit for the percentage of maximum torque.

Example:

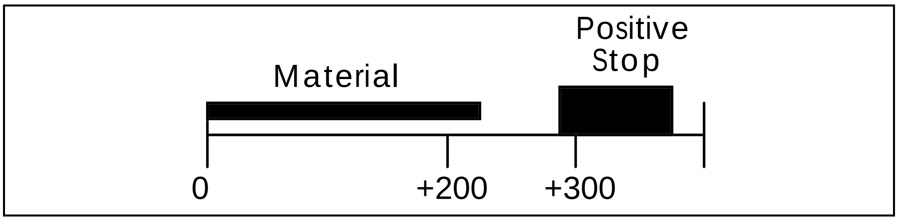


Figure 5-2: Positive Stop, MOM Command

```
000   JMP  0100
0100  MOM  1  020  030          ;Torque Reduction Before Stop = 20%
                                 At Stop      = 30%
0101  POA  1 +00250.000  300    ;Position Axis to 250 inches at 30%
                                 max vel.
0102  VCA  1 +00200.000  100    ;Velocity Change to 10% max. vel. at
                                 200 inches
0103  PFI  1 +00310.000  100    ;Position to positive stop= +310 inches
                                 at 10% max
0104  JMP  0200                 ;Jump to block 0200, Positive Stop
                                 Error Routine
0105  BCE  0120  02  1          ;Jump to block 0120, Input 2 is on,
                                 Return Home
0106  JMP  0105                 ;Jump to block 0105, Wait for Input 2
                                 to turn on
0120  PSA  1 +00000.000  999    ;Position axis to +0 position (Home) at
                                 99.9 % max.
0121  JST  0100                 ;Jump and stop at block 0100, wait for
                                 cycle start
0200  AEA  07  1                ;Turn on auxiliary output #7, Positive
                                 Stop Error
0201  PSA  1 +00000.000  999    ;Position axis to +0 position (Home) at
                                 99.9 % max.
0202  JST  0100                 ;Jump and stop at block 0100, wait for
                                 cycle start
```

M== Selection of an M function

```
E  1000 M==
mm
```

mm - Number (00...63) of the M function to be selected

A pre-coded M function can also be set in a separate block using “M==“.

If the M functions set are time dependent, it is important to note that their status will change according to the M function table after the set time in Parameter B009 has expired.

The status of the inputs and outputs, after processing the M function, are defined in the M Function table found in Section 4.6 of Chapter 4. The table of M functions is specified in parameters only and can be modified by the CTA keyboard or serially through the RS 232/RS 485 port.

NOP No Operation (Blank Block)

```
E  0021 NOP
```

This is the command to describe a blank block. During processing in the automatic mode, this blank block is scanned only: processing continues with the next block.

NOP's can be used to reserve program block space for future program change and expansion. Also use NOP's in the program where you may want to add commands "on the fly" to modify production.

NOP's may also be used as a program delay for each block. Note that the smallest delay programmable in the WAI command is 10 milliseconds. The NOP command allows for smaller increments of delay.

The default delay time for each NOP block is 1 millisecond.

PBK Positioning Interrupt

```
E  0055 PBK
1
```

1 - Axis 1

The PBK command stops the execution of the current motion profile. The axis is braked to a controlled stop. The axis will decelerate using the current programmed acceleration/deceleration rate. The remaining distance to be traveled is ignored. If continuous operation with CON was ON, it is switched OFF. After reading the PBK command, the program jumps to the next block.

This command could be used in another task to control the actions of the axis.

Note: The KDI command could be used for storing the approximate remaining distance, otherwise the information is permanently lost.

Example:

```
0000 CON 1 1 +999 00
0001 WAI 02.00
0002 PBK 1
0003 POI 1 +000050.00 100
```

After reading in the PBK command, the axis then drives the braking path from V=99.9% to V=0 plus 50 EGE units. However, a flowing transition from V=99.9% to V=10% [actually] results.

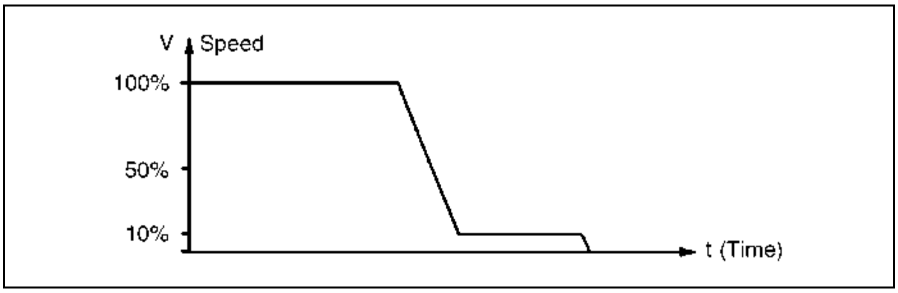


Figure 5-1: PBK Flowing Transition

PFA Absolute Position Feed To Positive Stop

```
E  1346 PFA
1 +00020.000 999
```

1	-	Axis 1
+00020.000	-	Absolute position feed to a positive stop in input units (to 2 or 3 decimal places, depending on the setting in B007)
999	-	feed rate in % (00.1-99.9) of the maximum velocity set in parameter (A100).

The PFA command is used in conjunction with the MOM command. The PFA command will absolute position the axis to a positive stop and the MOM command will determine the percentage of maximum torque during the move to the positive stop and the maximum torque at the positive stop.

When using the PFA command, the positive stop must be within the software overtravel limits, set in parameters A113 and A114. The absolute position, programmed in the PFA command, must be greater than or equal to the exact distance to the positive stop, otherwise the positive stop cannot be reached. The programmed absolute position can be greater than the software overtravel limits. Only one PFA command can be active at one time.

The program flow continues immediately after one or both of the following conditions:

- If positive stop is not reached, the next block in the program is executed. The next block is typically a unconditional jump (JMP) command to a error routine to notify the operator that the positive stop was not reached.
- If the positive is reached, the program flow will jump over the block immediately after the PFA command.

Refer to the MOM command in Chapter 5 for more information and an example of how the PFA command will operate.

Notes: 1. The axis must be homed prior to the execution of this command.

2. If the positive stop is not reached, the maximum torque of the axis and the overload factor of the drive are active.

Warning: When the MOM command is used and while positioning to a positive stop or positioned at the positive stop, the overload factor in the drive is internally set to 400 percent. The MOM command is the only limit for the percentage of maximum torque.

PFI Incremental Position Feed To Positive Stop

```
E  1347 PFI
1 +00020.000 999
```

1	-	Axis 1
+00020.000	-	Incremental position feed to a positive stop in input units (to 2 or 3 decimal places, depending on the setting in B007)
999	-	feed rate in % (00.1-99.9) of the maximum velocity set in parameter (A100).

The PFI command is used in conjunction with the MOM command. The PFI command will incremental position the axis to a positive stop and the MOM command will determine the percentage of maximum torque during the move to the positive stop and the maximum torque at the positive stop.

When using the PFI command, the positive stop must be within the software overtravel limits, set in parameters A113 and A114. The incremental position, programmed in the PFI command, must be greater than or equal to the exact distance to the positive stop, otherwise the positive stop cannot be reached. The programmed incremental position can be greater than the software overtravel limits. Only one PFI command can be active at one time.

The program flow continues immediately after one or both of the following conditions:

- If positive stop is not reached, the next block in the program is executed. The next block is typically a unconditional jump (JMP) command to a error routine to notify the operator that the positive stop was not reached.
- If the positive is reached, the program flow will jump over the block immediately after the PFI command.

Refer to the MOM command in Chapter 5 for more information and an example of how the PFI command will operate.

Note: If the positive stop is not reached, the maximum torque of the axis and the overload factor of the drive are active.

Warning: When the MOM command is used and while positioning to a positive stop or positioned at the positive stop, the overload factor in the drive is internally set to 400 percent. The MOM command is the only limit for the percentage of maximum torque.

P0A Position Absolute

```
E  2800 POA
1 +12345.678 999
```

1	-	Axis 1
+12345.678	-	Plus or minus absolute target position in input units (to 2 or 3 decimal places, depending on the setting in B007)
999	-	feed rate in % (00.1-99.9) of the maximum velocity set in parameter (A100).

The axis must be homed (G74, HOM or CLA) prior to execution of this command. Otherwise, an error message will be issued. Stepping to the next block takes place immediately after the absolute position has been read into the position buffer. Since this command does not wait for the axis to be in position before jumping to the next block, the program command following this command will be executed. This allows other functions to be executed while the axis is moving toward the end position.

If multiple POA commands are in sequence, the resulting move will, in most cases, continue and stop at the last POA Target Position issued.

An ATS command programmed in a subsequent block to monitor the Position Tolerance Output (Parameter A106) or the Position pre-signal (Parameter A107) is useful in controlling when to step to the next block.

POI Position Incremental

```
E  0290 POI
1 +12345.678 999
```

1 - Axis 1

+ - Incremental position direction (+ or -)

Note: If a character other than "+" is entered for this digit the direction will be minus.

12345.678 - Incremental target position in input units (to 2 or 3 decimal places, depending on setting in B007)

999 - Feed rate in % (00.1 - 99.9) of the maximum velocity set in parameter (A100).

Example:

500 would be 50.0% of the maximum velocity.

Stepping to the next block takes place immediately after the position block has been read in.

Since this command does not wait for the axis to be in position before jumping to the next block, the program command following this command will be executed. This allows the other functions to be executed while the axis is moving toward the end position.

If multiple POI commands are in sequence, the resulting move will be the sum of all incremental position commands.

The ATS command programmed to monitor the Position Tolerance Output (Parameter A106) or the Position pre-signal (Parameter A107) is useful in controlling when to step to the next block.

POM Position On Memory (Requires IDS Setup)

```
E  300  POM
1 0 0
```

1 - Axis 1

0 - Mode
0= Position incremental
1= Position in absolute dimensions

1 - Position direction
0= Positive
1= Negative

The POM command is an incremental or absolute positioning command. The programmed feed length and velocity are stored in a single memory location by an Indramat IDS decade switch option. Note that the axis can be controlled by the POM command, but given a single memory location, only one value can be stored at one time.

Stepping to the next block takes place immediately after the POM command is read into the program. This allows other functions to be executed while the axis is moving toward the end position.

The ATS command programmed to monitor the Position Tolerance Output (Parameter A106) or the Position pre-signal (Parameter A107) is useful in controlling when to step to the next block.

When using the IDS decade switch, enable this option in Parameter B003. If the decade switch option is not enabled, this block is scanned and no positioning takes place.

PSA Position Absolute (With In-Position Signal)

```
E  0310 PSA
1 +12345.678 999
```

1	-	Axis 1
+12345.678	-	Plus (+) or minus (-) absolute target position in input units (to 2 or 3 decimal places, depending on the setting in B007)
999	-	Feed rate as % (00.1 - 99.9) of the maximum velocity set in parameter (A100).

Note: The axis must be homed prior to execution of this command. Otherwise, an error message will be issued.

Stepping to the next block takes place only after the positioning has been completed. Parameter A106 (Position Tolerance) is used to set the position tolerance for stepping to the next block.

PSI Position Incremental (With In-Position Signal)

```
E  1420 PSI
1 +12345.678 999
```

1	-	Axis 1
+	-	Incremental position direction, plus (+) or minus (-)

Note: If a character other than "+" is entered for this position, the direction will be negative.

12345.678	-	Incremental target position in input units (to 2 or 3 decimal places, depending on setting in B007)
999	-	feed rate as % (00.1 - 99.9) of the maximum velocity set in parameter (A100)

Stepping to the next block takes place only after the positioning has been completed. Parameter A106 (Position Tolerance) is used to set the position tolerance for stepping to the next block.

PSM Position On Memory (with In-Position Signal) (Requires IDS Setup)

```
E  330  PSM
1 0 1
```

1 - Axis 1

0 - Mode

0= Position incremental

1= Position in absolute dimensions

1 - Position direction

0= Positive

1= Negative

The PSM command is an incremental or absolute positioning command with position acknowledgment. The programmed position distance and velocity are stored in a single memory location by an Indramat IDS decade switch option. Note that the axis can be controlled by the PSM command, but given a single memory location, only one value can be stored at one time.

Stepping to the next block takes place after positioning is complete. The "In Position" aux. output is programmed in Parameter A106 - Position Tolerance.

When using the IDS decade switch option, enable this option in Parameter B003. If the decade switch option is not enabled, this block is scanned and no positioning takes place.

PST Position Test

```
E  340  PST
1  05 +12345.678
```

1	-	Axis 1
05	-	Auxiliary output number
+12345.678	-	Test position value (to 2 or 3 decimal places, depending on setting in B007)

This command is used to check for a position and control an output based on current position status. At the execution of this block, the designated auxiliary output is turned ON if the actual position of the axis is less than the test position value.

If the actual position is equal to or greater than the test position value, the auxiliary output is turned OFF. Stepping to the next block takes place immediately after the block has been read in. Please note that the position is only tested once during the execution of the PST block. A program loop must be created to test for a position repeatedly.

Note: PST only functions after the axis has been homed. Prior to this, the block is scanned only and no execution takes place.

REF Referencing (Detect Registration Mark Input)

```
E  0350 REF
1 0 123 01
```

1	-	Axis 1
0	-	Direction 0= forward 1= reverse
123	-	Search speed in % (00.1 - 99.9) of the maximum velocity set in parameter (A100).
01	-	Registration auxiliary input number (Recognition in approx. 1 msec.)

It is possible at any time to search for a registration (reference) mark. The axis direction, search speed (velocity) and registration input number can be freely selected. After the start of the block, the axis moves at the specified search speed until a positive edge occurs at the specified aux. input. Stepping to the next block takes place only after the reference marker has been found (unless the next block is a REP command, see page on REP command).

When the registration mark is found, the axis decelerates and travels in reverse to the position the registration mark was detected at. If it is desired that the axis does not back up in this manner, the REF (or REF/REP) can be followed by a PSI command. The PSI position command would be set to the distance it takes to decelerate the axis from the search speed.

REP Conditional Jump if Search Distance is Exceeded, Referenced to REF Command

```
E  0030 REP
1000 1 123456.78
```

1000 - Target block number (0000-2999), if search path exceeded

1 - Axis 1

123456.78 - Maximum search path for REF command (to 2 or 3 decimal places, depending on setting in B007)

The REP command is a supplement to the REF command. It is used to limit the maximum search distance while looking for the registration input. If the maximum search distance entered here is exceeded without finding a registration input, a jump is executed to the target block. At the same time the drive is braked to a stop.

The REP command must follow immediately after the REF command. A REP command that stands alone leads to the error report “Illegal Command” during program execution.

At the start of the REF/REP commands, the axis moves at the search speed until either the registration mark is detected, or the search length has passed.

If the registration mark is not detected, the axis will decelerate to a stop after the search length occurs. The actual length traveled will be the search length plus the distance it takes to decelerate the axis to a stop. A jump is executed to the target block at the end of the search length, as the axis starts decelerating.

If the registration mark is detected, the axis is decelerated (as described for the REF command) and stepping to the next block takes place.

The target block should be the start of a recovery routine written to restore operation from a missed mark. It will need to be customized for each application.

The following command combinations are permitted:

1. Reference position travel, without driving to an offset dimension and without search distance limiting,

```
0020 REF 1 0 050 05
```

2. Reference position travel, driving to an offset dimension without search distance limiting,

```
0000 REF 1 0 50  10
0001 PSI 1 +000200.00 100
```

3. Reference position travel is programmed in the REF command. Limiting to max. 500 mm is programmed in the REP command.

```
0030 REF 1 0 100 12
0031 REP 0900 1 000500
```

4. Reference position travel with search distance limiting and offset dimension.

```
0055 REF 1 1 222 11
0056 REP 0900 1
0057 POI 1 +000123.00 99
```

Reference position travel is programmed in the REF command.

Limiting to max. 900 mm is programmed in the REP command.

An offset dimension of 123 mm is programmed in the POI command.

The REP command and the feed (POI or PSI possible) are processed in the same controller cycle as the REF command.

RMI Registration Mark Interrupt

```
E  0300 RMI
0 01
```

This command can be used to initiate additional processing, independently of the positioning program. The additional registration mark processing is inititated after the recognition of an input pulse, and after an offset has been traversed. The startup signal edge of the pulse is evaluated. The positioning program is interrupted for this registration mark processing and subsequently continued.

0 - Mode

- 0 = Wait for pulse at the registration mark input (stepping to the next block takes place after recognition of the pulse and traversing the offset.)
- 1 = Terminate registration mark processing continue program. (stepping to the next block takes place immediately)

01 - Registration mark input

- 00 = No registration mark selected
- 01-16 = Standard auxiliary input (recognition in approx. 1 msec)

The registration mark program has to be programmed in Task 2. An offset can be programmed in the form of an incremental position command (POI, PSI, POM, or PSM). The incremental position command has to follow the RMI command directly.

Example of application:

Perforated sheet metal should be cut according to the location of the perforations, which are stamped into the flat sheet metal on the stamping table. The material is then fed through the pass rolling mill. A hole serves as a reference signal for recognition of the registration mark. Since an unknown change of length occurs during profiling, the same feed program cannot be used for the cutting process because the possibility that the perforation will drift to the edge cannot be excluded. Because of this reason, cutting is done depending on the reference hole.

The reference hole can be detected by means of a light barrier, initiator, or similar device.

Notes:

- Cannot be used simultaneously with a REF command
- Will be ignored in the case of re-start
- The RMI works correctly only with feeds and offset feeds in positive direction (e.g., PSI 1 +....)
- In Task 2, (RMI program) there should be no other feed except for the offset
- After the RMI command (RMI 0 XX) has been called up, masking a hole is no longer possible
- Recognition of an additional registration mark prior to the command 'RMI 1 XX' is not possible

The velocity that can be changed by means of the offset feed is returned to its old value when the program resumes. The interruption, feed monitoring and override functions may be used.

Sample program:

```
Task1:      PSI 1 +000050.00 999
            JSR stamp1
            PSI 1 +000025.00 999
            JSR stamp2
            JMP task1

Stamp1:     AEA 01 1               ;stamping with stamp 1
            WAI 00.20
            AEA 01.00
            RTS

Stamp2:     AEA 02 1               ;stamping with stamp 2
            WAI 00.20
            AEA 02 0
            RTS

Task2:      RMI 0 01               ;wait for registration mark
                                   ;pulse and traversing of offset
                                   distance
            PSI 1 +000015.50 500   ;offset distance from position of
                                   registration mark

            JSR tool
            RMI 1 01
            JMP task2

Tool:       AEA 03 1               ;cutter on
            WAI 00.25              ;waiting time
            AEA 03 0               ;cutter off
            WAI 00.20              ;waiting time
            RTS
```

RSV Restart Vector

```
E  0700 RSV
1 000 10000
```

1 - Restore status as before interrupt, except outputs

000 - Not used, set to 000

1 - Outputs status

0 = Restore status of outputs as before interrupt

1 = Do not restore outputs

0000 - Not used, set to 0000

The Restart Vector is used to recover from a power loss, System Error or change in mode from Auto to Manual. When any of these events occur, the status at the time (velocity, absolute target position, outputs, etc.) are temporarily stored and can be recovered with this command. Use this command only if the system has an absolute encoder.

Note: The temporarily stored absolute target position will only be resumed if the axis is equipped with an absolute encoder.

RTS Return from Subroutine

```
E  0370 RTS
```

The RTS command is used to return from a subroutine which has been called by using the JSR command. The program is continued at the point where the subroutine had been called which is the next block after the JSR command.

If in multiple subroutines, the RTS command will take the program back only to the level of the current subroutine stack.

Note: Additional program flow control is possible by using the Change Subroutine Stack command (CST) with the JSR/RTS commands for advanced programming.

SAC Set Absolute Counter

```
E  0300 SAC
1 0   +12345.678
```

1 - Axis 1

0 - 0= Absolute offset (offset change)

1= Set absolute position with respect to the command position.

2= Set absolute position, with respect to the actual position.

+12345.678 - Absolute position or offset, plus (+) or minus (-)

This command is used to set or change the value of the absolute position counter. The axis must be homed before this command can be used.

SIN Sine Oscillation

```
E  0380 SIN
1 10 12.345  999
```

1	-	Axis 1
10	-	Auxiliary input which enables SIN function
12.345	-	Amplitude in Input Units (to 2 or 3 decimal places, depending on setting in B007)
999	-	Frequency of oscillation in Hertz (0.01 to 9.99)

This function will start up with 0 degrees when the enable input is switched On. When the enable input is switched Off, the function ends at 360 degrees. After this, stepping to the next block will take place. The axis must be homed, or an error will occur when this command is read.

S01 Scanning of Inputs and Modifying a Position/Velocity (Special Option #1)

```
E  1000 SO1
0 1 ee 0010
```

0 - 0 = Read decade switch values

1 = Convert input values into a position and store this in the position section of the POI, PSI, POA, PSA, VCC, PST, commands

2 = Convert input values into a velocity and store this in the velocity section of the POI, PSI, POA, PSA, VCC commands

3 = Convert into a length and write to the tool correction memory defined under "data target" replacing the information contained in this memory.

4 = Convert into a length and write to the tool correction memory defined under "data target", adding to the information contained in the selected memory.

5 = Convert into a length and write to the tool correction memory defined under "data target", subtracting from the information contained in the selected memory.

In modes 3 through 5 a maximum of 7 digits are evaluated (the 8th digit from the right is the mathematical operation).

1 - Decimal Position: To 2 or 3 decimal places, depending on setting in B007.

For 3 decimal places	For 2 decimal places
1= 10^{-3}	1= 10^{-2}
2= 10^{-2}	2= 10^{-1}
3= 10^{-1}	3= 10^{0}
4= 10^{0}	4= 10^{1}
5= 10^{1}	5= 10^{2}
6= 10^{2}	6= 10^{3}
7= 10^{3}	7= 10^{4}
8= 10^{4}	8= 10^{5}
9= Sign (+ / -)	9= Sign (+ / -)

ee - Starting input number for inputted data, with the BCD value:

ee= 10

ee+1= 11

ee+2= 12

ee+3= 13

0010 - Data target
In mode 1...2: Program block (0000 to 2999)
In mode 3...5: Number of the correction memory (01 to 19)

The SO1 command is used to read in length information via decade switches or from a programmable logic controller (PLC).

The information entered is cleared after a fault, Emergency Stop, when powering up the DLC and when entering Parameter Mode.

“Malfunction S01 Command” occurs if;

1) The Tool Correction register number is not between 01...19.

2) The amount of the correction value to be stored exceeds the value 99999.99 or 9999.999

In the following example, aux. input 04 is the least significant digit. Enter 04 for the ee entry. Input 05 is 04+1 (ee+1). Input 07 is the most significant digit (ee+3). The DLC must read in all decimal places, one after the other. A SO1 command is required for each decimal place. The resulting information is converted to position and velocity and stored in one of the following position commands (POI, PSI, POA, PSA, VCC) programmed in the target block.

Additional Examples:

A) Hardware Connect for an external decade switch array (4 decades).

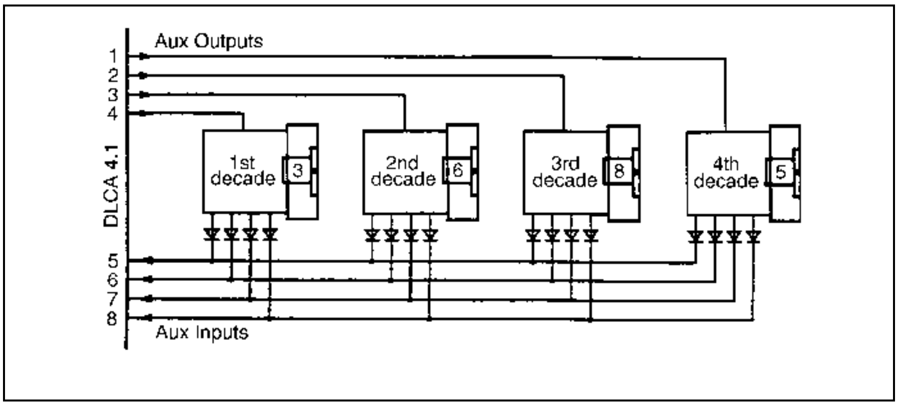


Figure 5-2: Decade Switch Array

The inputs are switched using the APE command.

The inputs are read using the SO1 command.

At the beginning of statement 0300 is POI 1 +123456.78 999.

B) Programming

0900 APE 0 0100022222 0901 WAI 00.02 0902 SOI 0 2 05 0000	-Select 4th decade -Wait until level at inputs is stable -Read in value of 4th decade
0903 APE 0 0010022222 0904 WAI 00.02 0905 SOI 0 3 05 0000	-Select 3rd decade -Wait until level at inputs is stable -Read in value of 3rd decade
0906 APE 0 0001022222 0907 WAI 00.02 0908 SOI 0 4 05 0000	-Select 2nd decade -Wait until level at inputs is stable -Read in value of 2nd decade
0909 APE 0 0000102222 0910 WAI 00.02 0911 SOI 0 5 05 0000	-Select 1st decade -Wait until level at inputs is stable -Read in value of 1st decade
0912 SOI 1 0 00 0300	-Convert information and send to target block
0913 JMP 0900	-Start new read-in cycle

After calling the program the statement 0300 contains:

0300 POI +000368.50 999

It is useful to create a program with SO1 commands in Task 3.

Notes:

1. It is useful to program this type of decade switch query in task 3.
2. All numbers which are not read in (pseudo-nibbles) are assumed to be 0.

STH Send to Host

```
E  0390 STH
0  0019
```

0 - Type of information

00 = Status 00, 02, 03, 05, 06, 07, 08, 09, 10, 19, 47, 48, 49, 50, 51, 52, 53

1 = Counter status of preset and actual number of pieces. The block number of the counter must be entered in the status code field.

0002 - Code for Status, options as below:

0000 =	Actual position of each axis	0041 =	Registration Position (REF)
0002 =	Current program block number	0043 =	Measuring Wheel
0003 =	Actual position as a hexadecimal value	0046 =	Length Counter
0005 =	Software version description	0047 =	Actual RPM, MW
x006 =	Current input state	0048 =	Actual Motor RPM
x007 =	Current output state	0050 =	System I/O (Hexadecimal)
0008 =	Current Block (Task 1, 2, 3)	0051 =	Aux. Inputs (Hexadecimal)
0009 =	Measuring Wheel	0052 =	Aux. Outputs (Hexadecimal)
0010 =	Following Error	0053 =	DLC System Error
0018 =	Servo Diagnostic		
0019 =	DLC Hardware / Software		

Note: In x06 and x07, x = Input/output bank number, 0-9

Program-controlled data output via the serial interface (see Chapter 7). Stepping to the next block takes place immediately after the block has been read in. Serial interface must have been activated through parameters B003 and B004. Otherwise, stepping to the next block takes place without having an output transmitted.

Examples of STH entry:

0__0003______

Status `03' actual position of axis 1 and 2 in Hexadecimal, is transmitted to Host device via the serial interface.

1__1000______

Status of counter located in block 1000. The desired and actual number of pieces, is transmitted to Host device via the serial interface.

STO Send Information To Outputs

```
E  0002 STO
101 1 3 0000 05
```

1 - Axis number or general data
  0 = Counter Status
  1 = Axis 1 Position

0 - Not used, set to 0

1 - Information Type
  1 = Absolute Position
  2 = Counter Status Of The BAC or COU Commands

1 - Mode
  0 = Store Information (position or counter)
  1 = Output Stored Information Via Outputs In BCD Format

3 - Decimal Position

Decimal	Significance
1	1st digit starting from the right
2	2nd digit starting from the right
3	3th digit starting from the right
4	4th digit starting from the right
5	5th digit starting from the right
6	6th digit starting from the right
7	7th digit starting from the right
8	8th digit starting from the right
9	9th digit is prefix; + = 0 - = 1

For the prefix, 24V is output for minus, and 0V is output for plus.

0000 - For Counter Status Information Type Only - Counter Command Block Number

10 - Starting Output Number XX, with the BCD value (see illustration):

XX	= 10	corresponding value	$1 = 2^0$
XX+1	= 11	corresponding value	$2 = 2^1$
XX+2	= 12	corresponding value	$4 = 2^2$
XX+3	= 13	corresponding value	$8 = 2^3$

The STO command is used to output either axis position or counter status information via four auxiliary outputs in BCD format. Only one type of information can be stored internally at one time, either absolute position or counter status. The user specifies the type of information, number of decimal places, counter block (if counter status is selected), and the starting output number. The three outputs after the starting output number will be used to output the information. The user program must be written to store the absolute position or counter status and then output them in the proper order.

STO Command Example:

For this example, the absolute position for the axis is equal to +598.00 inches. The auxiliary outputs 04 through 07 are used for the BCD value.

```
0900   AKN   02  0                ; Toggle input #2 to begin the program
0901   AKN   02  1
0902   STO   1  1  0 0 0000 00    ;Store actual position
0903   JSR   0925                 ;Jump to subroutine at program block 0925
0904   STO   1  1  1 9 0000 04    ;Output the direction of the sign
0903   JSR   0925                 ;Jump to subroutine at program block 0925
0906   STO   1  1  1 3 0000 04    ;Output third digit
                                  ;( XX+3=1, XX+2=0, XX+1=0, XX=0) = 8
0907   JSR   0925                 ;Jump to subroutine at program block 0925
0908   STO   1  1  1 4 0000 04    ;Output fourth digit
                                  ;( XX+3=1, XX+2=0, XX+1=0, XX=1) = 9
0909   JSR   0925                 ;Jump to subroutine at program block 0925
0910   STO   1  1  1 5 0000 04    ;Output fifth digit
                                  ;( XX+3=0, XX+2=1, XX+0 =0, XX=1) = 5
0911   JSR   0925                 ;Jump to subroutine at program block 0925
0912   STO   1  1  1 6 0000 04    ;Output sixth digit
                                  ;( XX+3=0, XX+2=0, XX+0 =0, XX=0) = 0
0913   JMP   0900                 ;Jump to program block 0900 and start a new read in the
                                  ;cycle
0925   AKN   01  0                ;Toggle input #1 to output the next digit
0926   AKN   01  1
0927   RTS                        ;Return from subroutine
```

VCA Velocity Change Absolute

```
E  0410 VCA
1 +12345.678 999
```

1	-	Axis 1
+12345.678	-	Absolute position to be reached before the velocity change (to 2 or 3 decimal places, depending on the setting in B007)
999	-	Speed in % (00.1 - 99.9) of the maximum velocity set in parameter (A100)

The VCA command is used to alter the current rate of velocity at some point along a path. Multiple VCA commands can be used to create a "step" effect in the velocity profile. Stepping to the next block takes place when the absolute position has been reached.

This function is designed to work in conjunction with the position commands POA, POI, and POM.

Example:

0000	POA	1	+00100.000	999	;Position axis to +100 inches at 99.9 % max.
0001	VCA	1	+00050.000	250	;Velocity change at +50 inches to 25.0% max.
0002	VCA	1	+00075.000	500	;Velocity change at +75 inches to 50.0% max.
0003	VCA	1	+00090.000	100	;Velocity change at +90 inches to 10.0% max.
0004	ATS	15	1		;Acknowledge axis is in-position (A106)
0005	WAI	01 . 00			;Dwell for 1 second
0006	JST	0000			;Jump and stop at program block 0000

VCC Velocity Change Command

```
E  0411 VCC
1  12345.678 999
```

1	-	Axis 1
12345.678	-	Path to be traversed before the velocity change (to 2 or 3 decimal places, depending on the setting in B007)
999	-	Speed in % (00.1 - 99.9) of the maximum velocity set in parameter (A100)

The VCC command is used to alter the current rate of velocity at some point along a path. Multiple VCC commands can be used to create a "step" effect in the velocity profile. Stepping to the next block takes place after the path specified in the VCC command has been traversed.

This function is designed to work in conjunction with the position commands POA, POI, and POM.

Note: If the path specified here is longer than the previous position command, the DLC will step to the next block after the axis reaches the position tolerance (see Parameter A106).

Example:

0000	POI	1	+00100.000	999	;Move axis +100 inches at 99.9% max.
0001	VCC	1	00050.000	250	;Velocity Change after 50 inches to 25.0% max.
0002	VCC	1	00020.000	500	;Velocity Change after 20 more inches to 50.0% max.
0003	VCC	1	00010.000	100	;Velocity Change after 10 more inches to 10.0% max.
0004	ATS	15	1		;Acknowledge the axis is in-position (A106)
0005	WAI	01 . 00			;Dwell for 1 second
0006	JST	0000			;Jump and stop at program block 0000

VEO Velocity Override Command

```
E  0035 VEO
1 2 3 456 7
```

1 - Axis 1

2 - Override input selection

0 = switched OFF or override in accordance with Parameter B013

2 = Override via Binary value at Inputs #1-7, Significance = input 1:20, input 7:27

3 = Override via value in Gray Code in inputs #4-7 (as in B013)

4 = Programmed Override (requires entry in 456-override velocity factor)

5 = Override via measuring wheel encoder 1 pulse frequency

3 - Override update

0 = Read the Override input value every program cycle

1 = Reads Override value only once (each time the command is executed)

456 - Override velocity factor

Velocity factor to 3 decimal places (.001-.999). This factor is significant only if '4' is selected in the second digit. Subsequent feeds are reduced by this factor.

999 = 99.9%

050 = 5.0%

7 - Function

0 = Override as factor - Override input value represents a factor to determine the resulting velocity. All subsequent feeds will be scaled.

1 = Override as limiting - Override input value represents a limiting factor. Position commands that have a lower velocity than this are not affected.

Activation of an override function using the VEO command has precedence over any activation in parameter B013.

The Binary or Gray scale might be used in conjunction with a PLC.

The limiting sector is typically a internal software switch in the user program.

Programmed override is a means of scaling or limiting the velocity of the subsequent process after the command is read and will remain in effect until it is changed.

The Override via measuring wheel encoder 1 pulse frequency can be activated only if the axis in parameter A123 is configured as a “Normal Feed Axis”. The measuring encoder 1 must be configured correctly in parameters B016 through B019. Otherwise, indexing to the next statement is executed. The feed constant and number of lines per revolution are calculated so the maximum axis velocity is equal to the measuring wheel velocity. This velocity can be adjusted by using the Follow command (FOL). If the axis achieves a velocity greater than 1.25 times the axis maximum velocity, the error message "Override A1" will be displayed

The VEO command is used to override programmed velocity for all programmed commands.

1. With the function “Override as factor” the override velocity factor is multiplied by the programmed speed of the commands.
2. With the function “Override as limiting” the override is multiplied by the programmed speed parameter Vmax (parameter A100). This multiplied value is then valid as the max. speed limit.

WAI Wait (Time Delay)

```
E  0420 WAI
00.50
```

00.50 - Dwell time (00.01-99.99 seconds)

Examples of how a time delay is programmed:

00.01 = 10 milliseconds (the minimum dwell time programmable),

00.50 = 0.5 seconds.

The DLC waits in the program block until the specified time has elapsed. After the time delay, the program steps to the next block. See the NOP command for related function (when require less than 10 milliseconds dwell time).

WRI Write in Absolute Position (Teach Command)

```
E  0042 WRI
1234  001
```

1234	-	Target block number (0000-2999)
00	-	Command type
		00 = POA command
		01= PSA command
1	-	Axis 1

Use this command to "write in" (teach) an absolute positioning command in the user program. Note that an absolute positioning command requires that the axis must be homed. Otherwise, an "Axis 1 Not Homed" error occurs.

The entered target block number specifies where to install the absolute positioning (feed) command(s). The next entry digits specify which absolute position command is to be programmed in the target block.

POA = Absolute positioning command, stepping to the next block occurs immediately after it is read in.

PSA = Absolute position command, stepping to the next block occurs after positioning is completed.

This command is used to teach or update a position into a functional program. The axis is typically jogged into a desired position, then a predefined teach routine is run that will teach this position into all the appropriate blocks. The teach routine can be written as a manual vector or as program for Task 1, 2 or 3.

If the command stored at the selected target block is not a position command, the WRI command will write the absolute position command over the existing command stored in the block.

The absolute position command selected by the WRI command must have a distance to feed. This distance is taken from the Actual Position Display**. The velocity automatically inserted for doing the positioning is always the maximum (99.9%).** A preceding VEO (selection 4) or subsequent VCC (zero feed length) can be used to reduce the written velocity to a lower percentage.

Using the WRI command in Manual Mode

This is an example of how to "write in" absolute positions in the main program.

Task 3 is selected as the location for the WRI command. The Task 3 program must be loaded into the DLC before enabling parameter B006. Task 3 is always running, even before Cycle Start is pressed.

Start Task 2&3

Parameter B006   00000150

Note: Axis must be homed.

Example:

Program for the WRI command.

Block #			
0150	AKN	02 1	;Aux input #2 "writes in" 1st position
0151	BCA	0150 15 0	;Check axis is homed
0152	WRI	0015 001	;Writes in Abs. Position at Block 0015
0153	AKN	02 0	;Write-in complete
0154	AKN	02 1	;Aux. Input #2 "writes in" 2nd position
0155	WRI	0019 001	;Writes in Abs. Position at Block 0019
0156	AKN	02 0	;Write-in complete
0157	JMP	0150	;Jump to Block 0150

Example:

Main Program Before It "Writes In" the New Position

Block#		
0000	JMP	0015
0015	PSA	1 +00000.000 500
0016	AEA	01 1
0017	WAI	00.50
0018	AEA	01 0
0019	PSA	1 +00000.000 500
0020	AEA	02 1
0021	WAI	00.50
0022	AEA	02 0
0023	JMP	0015

Procedure for Using the WRI Command in Manual Mode

1. In Manual mode, home the axis.
2. Jog the axis to the first position desired.

Example:

First position desired: L P 1A +00010.000

1. Turn Auxiliary Input #2 on and off.
2. Jog the axis to the second position desired.

Example:

Second position desired: L P 1A -00005.000

1. Turn Auxiliary Input #2 on and off.
2. Now blocks 15 should change to the first position desired. Blocks 19 should change to the second position desired.

Example:

Main Program After It "Writes In" the New Position Block Number

0015	POA	1 + 00010.000 999	;Always writes in velocity of 99.9%.
0016	AEA	01 1	;Use VEO or VCC to change it.
0017	WAI	00.50	
0018	AEA	01 0	
0019	POA	1 - 00005.000 999	;Always writes in velocity of 99.9%.
0020	AEA	02 1	;Use VEO or VCC to change it.
0021	WAI	00.50	
0022	AEA	02 0	
0023		JMP	0015

5.10 Tool Correction

D==, G40, G43, G44

```
Tool Override

Dnn ±00010.00
```

nn	-	Number of the Tool Correction Register to be used.
00000.00	-	Tool Correction value (to 2 or 3 decimal places, dependent on the value in parameter B003). A sign bit must be added. Positive correction is the default.

Input

The tool length correction consists of a tool correction memory with 19 selectable Tool Correction registers, D01...D19. D00 is effective upon entering automatic operation and is active once the program is started.

Inputting of correction values is possible on-line through the CTA keyboard or through the serial interfaces. Tool Correction register values can be entered in manual and automatic mode. External Tool Correction data input is also possible through parallel inputs using the SO1 command (see Chapter 5 for a complete description of the SO1 command).

It is possible to use the Tool Correction register for a variable memory, however, only one correction value can be selected and calculated per programming command.

Each correction value memory can accept a seven-digit correction value. The eighth digit is the mathematical operator (+/-).

The commands G43 and G44 enable the selected Tool Correction register. Its value is then added or subtracted from the destination programmed according to the mathematical operator (+/-).

Tool correction is disabled with the command G40 or by selecting the register D00.

Selecting a Correction Memory

D==

```
E 1000 D==
nn
```

nn - Number of the correct memory selected (00...19)

An activated tool correction is effective for **all** feed commands (G01, POA, POI, POM, PSA, PSI, and PSM).

Registers D01-D19 can contain values up to seven digits. The eight digit is designated as the sign digit, either + or -.

Register D00 deactivates Tool Correction and unselects any active register. After a D==00 block, in order to activate Tool Correction, you must re-select a Tool Correction register using the D==nn command.

Activating Tool Correction

G43 Tool Correction - Activated

```
E 1000 G43
```

Adds the contents of the active Tool Correction register to the programmed destination. The sign of the value in the Tool Correction register is considered in this computation.

Example:

If the value in the register is +1, this command will add +1 to the programmed destination. If the value is -1, this command will add -1 to the programmed destination.

G44 Tool Correction - Subtracted

```
E 1000 G44
```

Subtracts the contents of the active Tool Correction register to the programmed destination. The sign of the value in the Tool Correction register is considered in this computation.

Example:

If the value in the register is +1, this command will subtract +1 to the programmed destination. If the value is -1, this command will subtract -1 to the programmed destination (which means it will add 1 to the programmed destination).

Deactivating Tool Correction

There are two ways to deactivate a tool correction value:

- Select Tool Correction memory 00 with a D==00 command.
- Disable the Tool Correction function with the G40 command.

G40 Tool Correction Deactivated

```
E 1000 G40
```

This command disables Tool Correction within the process path in the programmed feed command.

G40 is effective upon entering automatic operation.

The register selected remains selected, even after deactivating the Tool Correction with G40. If a G43 or G44 command is issued after a G40 is issued, the last selected Tool Correction register prior to the G40 command will be used as the selected value.

External Tool Correction

Tool Correction registers can also be specified on-line using the SO1 command. The SO1 command is used to read in information via decade switches or from a programmable controller.

6 Installation/Start-Up

This Chapter covers the installation and initial start up testing procedures for the DLC control system. The typical DLC Control system consists of the following components:

- DLC Control Module Card
- DEA 4.1 System/Auxiliary I/O card
- CTA Keypad & Display with remote mounting kit(Optional) and IKS745 cable
- Power Supply
- Digital AC Servo Drive
- MDD Digital AC Servo Motor

The instructions in this chapter primarily describe the installation of the DLC Control Module Card. The installation particulars of the other components (such as a TVD, TVM or KDV power supply, the DDS Digital AC Servo Drive, or DKS self-contained power supply and digital drive module) are explained in detail in their respective manuals.

Tools required:

- A small 1/8" blade, standard screwdriver
- A multimeter (VOM)
- An appropriate wrench for 1/4" machine type bolts (used for mounting module(s) to the cabinet.
- VT 100 terminal or PC with VT 100 terminal emulation software
- IN 391 Cable, PC to Indramat Digital AC Servo Drive

6.1 Mounting Cabinet

It is recommended the Amplifier and Power Supply be mounted in a cabinet, side by side (see Figure 6.1). The enclosure should be sufficient to protect the equipment from contaminants, such as water, oil, etc. Indramat recommends a NEMA 4 or 12 enclosure or equivalent.

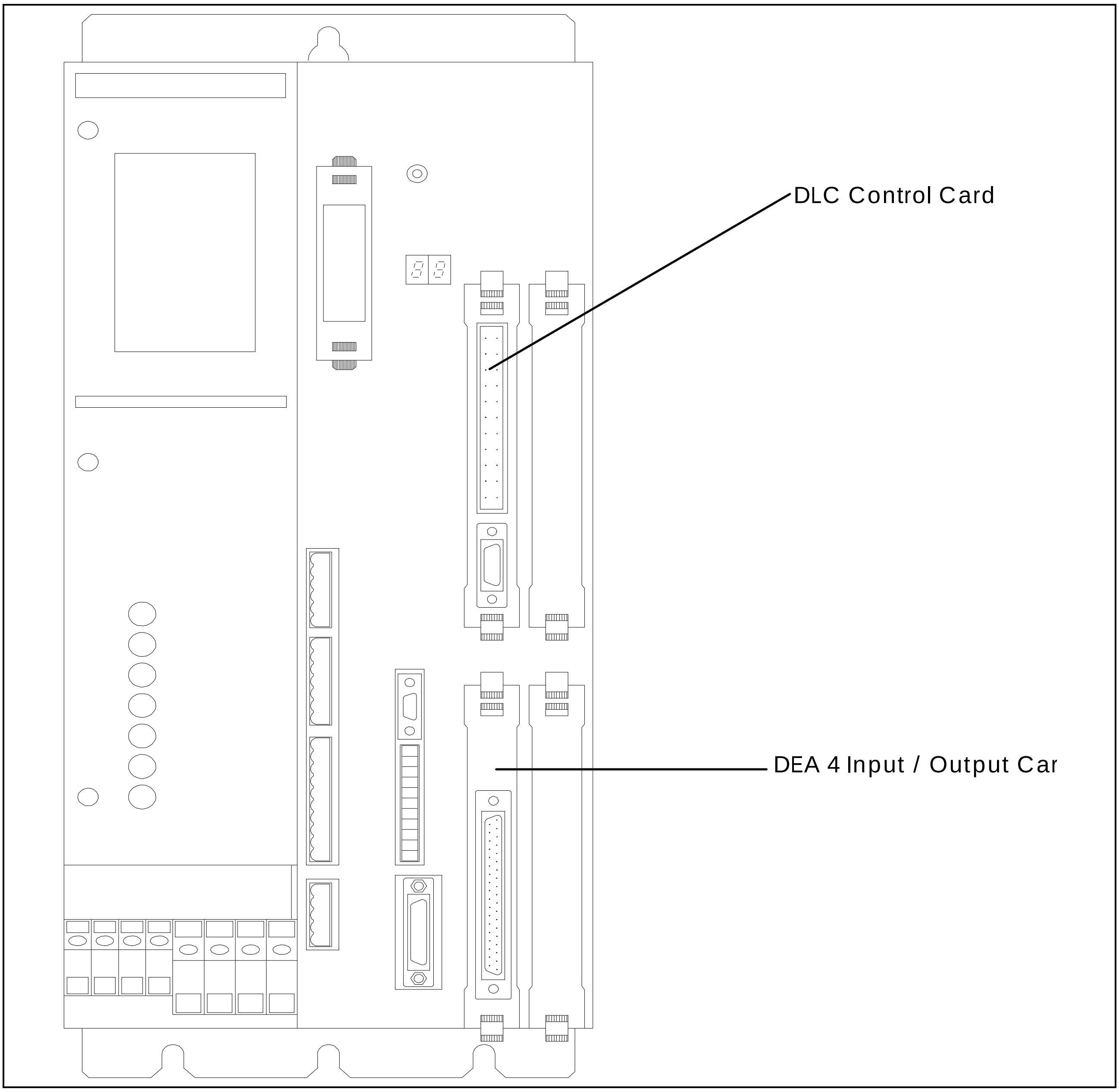


Fig. 6-1: DKS/DLC Servo System with DEA 4 Installed

6.2 Power

The DLC receives its power from the digital drive being used. Refer to separate manuals (listed in Appendix B) for power requirements of the other modules (power supply, servo drive, etc.).

6.3 Cable Routing

For higher reliability, the high resolution feedback cables must be shielded and routed away from high voltage power sources to reduce electrical noise.

Do not route the cables near high amperage type machines, like welding equipment, which produce strong magnetic field interference.

Suppress inductive loads (such as solenoids and motors) that are switched ON and OFF during DLC system operation, with R-C networks (AC) or diodes (DC).

Correct grounding is essential for trouble free operation. The ground connection must be made by observing strictly the branching conditions shown in the wiring diagram.

1. Transformer - Heat Dissipation

The incoming 3-Phase power must be ground referenced. Use an isolation transformer with a "Y" secondary if this cannot be confirmed. Refer to TVD/TVM/KDV/DKS (or appropriate) manual for additional information.

If a transformer is needed for operation, exercise caution on its location. Do not install the transformer in the same cabinet as the modules unless a sufficient method of cooling is applied.

6.4 Hardware Installation

See Appendix D for mounting hole dimensions and location of the CTA keypad and display module, the cut-out dimensions for the remote keypad, keypad gasket and mounting holes and cut-out dimensions for the optional IDS.

The mounting dimensions and cabinet cut-out data for the TVD/DDS or DKS are located in the units respective support manuals.

The method of mounting the MDD Digital AC servomotor is dependent upon the application. Additional information and a drawing package for the MDD servomotor is available on request from Indramat.

6.5 Electrical Installation

There are many variations of wiring techniques used to connect the DEA 4 System/Auxiliary I/O card to the machine builder's equipment. The Interconnect Drawings supplied with this manual (see Appendix C) show an example of the connections to a typical machine.

6.6 DLC/DEA 4.1 Connectors

This section describes the connectors on the DLC/DEA 4.1 plug-in cards. Refer to Figure 1.5 in Chapter 1 for an illustration of the DLC/DEA 4.1 card and its connector locations in the DKS.

X17: DEA 4.1 System/Auxiliary Input/Output Cable (Cable 04-0123)

X30: CTA Keypad & Display

Female, 34 pin shielded ribbon cable connected to the front panel of the DLC. The cable is an IKS 745 and its standard length is 2 meters. It connects from the DLC to the back of the CTA. Refer to the DKS/DLC interconnects in Appendix C for the proper orientation of the IKS 745 cable.

X31: RS-232/485 Communications Port (Interface)

Female, DB 9 connector is located on the lower front panel of the DLC. It is used for serial communication between the DLC and a host terminal, SOT or the IDS option. Refer to Chapter 7 for description of this multi-functional, two-way communications port.

6.7 Pre-Operation Start Up Tests

The following sections are intended to provide the user with an example of a DLC Digital AC Servo System start-up. They provide an example of a single axis application that will verify proper function of the system.

WARNING: The information given in the following sections may not be suitable for a specific application. If you use this example for testing, do not mechanically connect the servomotor to the actual load.

6.8 Connections

Note: Do **not** apply power until all connections have been made. Refer to the drawings in Appendixes C and D for mounting and connection information

1. Install the TVD/DDS (Power Supply/Digital AC Servo Drive or DKS (Digital Compact Drive) per their respective support manuals.
2. Connect the MDD motor power cable and high resolution feedback cable to the DDS or DKS per interconnect drawings in Appendix C, or refer to the DDS or DKS support manuals.
3. Connect the optional CTA Keypad/Display to the DLC Connector X30 via IKS 745 ribbon cable. Refer to the interconnect drawings in Appendix C.

6.9 DEA 4.1 Input Connections

The DEA 4.1 input connections used in this example are minimized for the sake of simplicity. Refer to the interconnect drawings in Appendix C for any additional details. Connect the incoming side of each switch to a external +24 Vdc source.

1. Connect a 2-position selector switch to provide a +24 Vdc signal to Pin 2 (Automatic Mode input). The second position is not wired (0 Vdc), and is used for Manual Mode (default mode). Connect a key switch to provide +24 Vdc to Pin 1 (Parameter Mode input).
2. Connect a Normally Closed pushbutton switch to Pin 3 ($\overline{\text{Emergency Stop}}$) of the DEA 4.1, connector X17.
3. Connect a Normally Open momentary pushbutton to Pin 4 (Cycle Start) of DEA 4.1, connector X17.
4. Connect a Normally Closed momentary pushbutton to Pin 5 ($\overline{\text{Cycle Stop}}$) of DEA 4.1, connector X17.
5. Connect a Normally Open momentary pushbutton to Pin 6 (Jog Forward) of DEA 4.1, connector X17.
6. Connect a Normally Open momentary pushbutton to Pin 7 (Jog Reverse) of DEA 4.1, connector X17.
7. Connect a Normally Open momentary pushbutton to Pin 8 (Clear) of DEA 4.1, connector X17.
8. Connect the +24Vdc external power supply to pins 34 and 35 (0Vdc) and pins 36 and 37 (+24Vdc) of the DEA 4.1 connector X17.

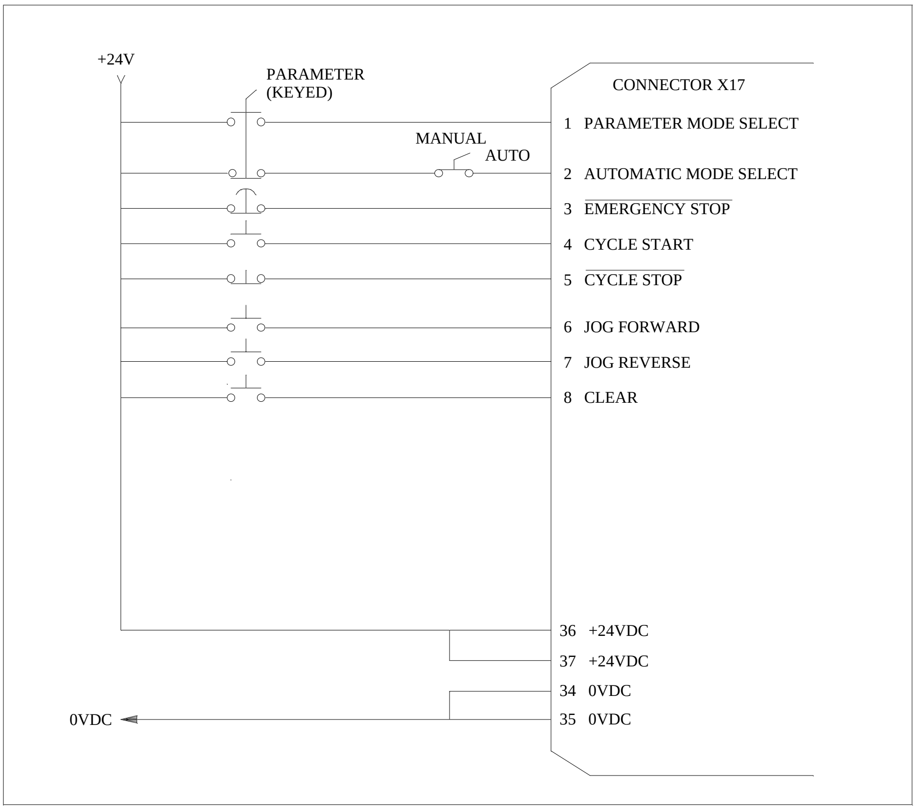


Fig. 6-2: Example Input Diagram

6.10 DEA 4.1 Output Connections

For the purpose of a sample start-up, the connections for the DEA 4.1 outputs are not necessary. The ends of the unused wires in cable 04-0123 should be wrapped in electrical tape to prevent any damage to the outputs. Refer to the interconnect drawings in Appendix C for properly interfacing the DEA 4.1 system /auxiliary outputs with the machine.

6.11 Power-Up

The system can now be powered up. All voltages should be checked by a qualified electrician to ensure proper signals and connections.

The digital drive being used has two 7 segment LEDs, H1, which displays the current operating status of the digital controller. If all connections are correct the H1 status will be "bb". If the diagnostic is not "bb", consult the digital controller manual for information on how to correct the problem.

The CTA display should also display no errors. Consult Chapter 8, Diagnostics And Troubleshooting for more information on how to correct the problem.

The DLC Digital Servo Systems is ready for drive & system parameters to be installed.

6.12 Digital AC Servo Amplifier Parameter Entry

The Indramat Digital AC Servo Amplifier contains an RS232 interface, Connector X2. This interface gives you information regarding the status of the drive and allows you to read and adjust all the amplifier's system parameters. The RS232 interface provides screens and menus which are accessible via a VT 100 terminal or a PC computer with a VT100 emulator, (e.g. Procomm, Windows 3.1, etc.). The communication settings are as follows:

Baud Rate:	9600
Parity:	None
Data Bits:	8
Stop Bits:	1

Connect the VT 100 terminal or PC with VT 100 terminal emulation software to the Digital AC Servo Amplifier to connector X2 via an IN 391 cable and press the <Enter> key. This will bring up the Main Menu screen.

Note: Digital drive parameters can also be set with the CTA keypad. If using the keypad for drive parameter entry, skip this section and refer to Chapter 4.

Changing The Display Language

In order to change the language displayed on the screen, press the number 3 from the Main Menu.

German Text = Press the number 1

English Text = Press the number 2

After the selection, the Main Menu screen will be displayed.

Main Menu

The Main Menu allows you to view the following information:

- Drive Status
- Parameters
- Language Selection
- ESC = Exit Program

In order for the DLC to operate properly, the Digital Amplifier's parameters must coincide with the parameters programmed in the DLC. To check or adjust the parameter currently in the amplifier, press number 2 from the Main Menu screen.

The Drive Parametrization screen will be displayed. Consult the amplifier's manual for a description and the proper parameter values for the amplifier/motor combination being used, press the letter “c” to continue.

The Parameter Menu screen will display the following selection list, application parameters, drive parameters, and parameter management. When setting up an Indramat Digital Amplifier with the DLC, the application and drive parameters must be set correctly.

Setting the Amplifier's Parameters to Operate with the DLC Control Card

1. Press the number 6, Motor & Drive Data.
   This will display the current amplifier/motor combination. Look under the Motor Parameters and make a note of the Maximum Velocity of Motor (RPM). Press <ESC> to return to the Parameter Menu.
2. Press the number 1, Operation Mode Scaling.
   The Operation Mode allows the user to configure the following amplifier parameters:
   - Analog Outputs
   - Bipolar Velocity Limit Value
   - Overload Factor
   - Analog Output - Position Data Scaling
   - Analog Output - Velocity Data Scaling

Analog Outputs

The Indramat Digital Amplifier has two analog outputs. AK1 (Connector X3, pin 1, referenced to OVm, pin 2) and AK2 (Connector X3, referenced to Ovm, pin 4). These outputs can be configured to output following items:

- Velocity Command Value
- Velocity Feedback Value
- Position Feedback Value
- Motor Encoder - Sine
- Motor Encoder - Cosine
- Current Command Value

Press the up or down arrow keys to position the cursor on the proper channel. Then press the left or right arrow keys to select the desired output. Continue pressing until the desired output is selected.

Bipolar Velocity Limit Value

The Bipolar Velocity Limit Value sets the motor's maximum velocity limit in the forward or reverse direction.

Position the cursor to the right of the Bipolar Velocity Limit Value and type in the desired velocity in revolutions per minute (RPM).

Note: The value entered should not exceed the maximum velocity of the motor. If an operating velocity of less than maximum is desired, enter that value here.

Overload Factor

The Overload Factor is determined based on the desired intermittent torque and duty cycle of the system. Consult the Indramat Digital Amplifier Selection List for the proper Overload Factor Value.

For the purpose of testing the DLC with the Indramat Amplifier, set the Overload Factor to 100%. Position the cursor to the right of the Overload Factor and type 100, then hit enter.

Position Data Scaling at the Analog Output (Degrees/10V)

When the position values are selected for either of the analog outputs, the Position Data Scaling defines a zero to ten volt linear scaling based upon the angle of the motor's rotor position.

For example, if 360 degrees were entered, one revolution of the motor would provide one linear voltage ramp from 0 to 10V.

Position the cursor to the right of the Position Data Scaling and type in the desired degrees. Typically, 360 degrees is entered.

Velocity Data Scaling At The Analog Output (RPM/10V)

When the velocity values are selected for either of the analog outputs, the Velocity Data Scaling defines a zero to ten volts linear scaling based upon the velocity of the motor in RPM. For example, if the Bipolar Velocity Limit is 3000 RPM, then when the motor is operating at 1500 RPM, 5 volts will be the output.

1. Position the cursor to the right of the Velocity Data Scaling and type in the desired velocity per 10 Volts.
   Typically the value entered here is the same value entered in the Bipolar Velocity Limit Value.
2. Press <ESC> to return to the Parameter Menu.
   The Absolute Encoder Parameters are automatically set by the amplifier.
   When power is applied, the amplifier reads the type of motor connected and sets the Absolute Encoder Parameters based upon a single or multi-turn feedback.
3. Press the number 3, error Reaction.
   There are three types of Error Reaction to select. The Error Reaction selected determines how the amplifier will react when a shutdown error occurs. Typically 0 is selected as the Error Reaction.
4. Press <ESC> to return to Parameter Menu.
   The Torque Limits are currently not available.
5. Press the number 5, Gain Parameters.
   The Gain Parameters allow the user to adjust the velocity/current loop gains of the amplifier.
6. To set the amplifier/motor combination to standard values, position the cursor to Reset To Standard Parameters and press the <Enter> key.

7. The following question will be displayed: Are You sure? Yes - > Y,
8. Press the <Y> key on your computer to reset to standard parameters. Press any other key to prevent resetting to standard parameters.

Note: You must be in parameter mode, RF disabled, to reset to standard parameters.

9. Press <ESC> to return to the Parameter Menu.
   The amplifier's parameters entry has been completed. The parameter management section allows the user to save and load the parameters to and from diskette. Consult the Indramat Digital Amplifier Manual for more information.

The next step is to configure the DLC Control Card to operate with the parameters entered in the digital amplifier.

6.13 Parameter Entry

The parameters given here are the minimum required to operate the digital drive with the DLC control card. For this example, the following descriptions assume an input unit (IU) = 1 inch. Turn the keyswitch to Parameter Mode to enable parameter entry. Enter the following parameters (refer to Chapter 4 for parameter entry procedures or for further description of each parameter, if necessary).

B007 Language
01 03 0 0 00 (English display, resolution to 3 places)

A100 Max Velocity
See example entries on the following page or refer to Chapter 4, Parameters for more information.

A101 Jog Velocity
00005.000 (5.0 in/sec)

A102 Acceleration
0000025.0 (25.0 in/sec^2)

A103 Position Gain
0000 01.00 (1.0 in/min/mil of Following Error)

A106 Position Tolerance
01 000.050 (Output 01, position tol.= 0.050 in)

A108 Feed Constant
001.00000 (1.0 in/motor rev)

A109 Direction
0 0 000000 (Direction of operation is unchanged)

A110 Homing Setup
0 0 0 2 00 0 0 (Required for MDD High Resolution Multi-turn Feedback)

A121 Maximum Motor Speed
Enter maximum motor RPM here.

A122 Monitor Window
0 0 00 0 010 (10% window)

Following are three examples for A100 and A121 input. Refer to Parameter A100 description to calculate the Max Velocity.

Maximum Motor Speed	A121 Entry	A100 Entry
1500 rpm	1500 0000	00025.000 (25 in/sec)
2000 rpm	2000 0000	00033.333 (33.333 in/sec)
3000 rpm	3000 0000	00050.000 (50 in/sec)

6.14 Program Entry

This is a sample program for operating the DLC in an automatic mode. Enter the program as shown.

Block 0000

```
E  0000 PSI
1 +00100.000 200
```

Block 0001

```
E  0001 WAI
01.00
```

Block 0002

```
E  0002 PSI
1 -00100.000 200
```

Block 0003

```
E  0003 WAI
01.00
```

Block 0004

```
E  0004 JST
0000
```

Block 0000 - will command the axis to rotate in the plus direction 100 input units, at 20% of the maximum velocity. Since the feed constant equals 1, the motor should turn 100 revolutions.

Block 0001 - will cause a 1 second delay once the axis has reached position.

Block 0002 - will cause the axis to rotate in the minus direction 100 input units, at 20% of the maximum velocity.

Block 0003 - will cause a 1 second delay once the axis has reached position.

Block 0004 - will jump back to block 0000 and stop. See the following sections for testing procedures.

6.15 Axis Jogging in Manual Mode

With "bb" displayed on the digital drive and no errors displayed on the CTA, press the Drive On pushbutton. The H1 status should display "AF" and torque should be applied to the MDD servomotor. If problems exist, consult the digital drive manual or Chapter 8 of this manual, to correct the problems.

To verify proper motor hook-up, and control of the motor, jog the axis in Manual Mode. Turn the selector switches to the OFF position so that neither the Automatic nor the Parameter Mode inputs are high (defaults to Manual Mode).

1. Press the Forward Jog pushbutton (connected to X17, Pin 6). The axis should jog in the forward direction, until the button is released.
2. Press the Reverse Jog pushbutton (connected to X17, Pin 7). The axis should jog in the reverse direction, until the button is released.
3. Should either the forward or reverse motor movement fail to react, check all cable connections and verify the DLC is in Manual Mode.

Note: If the CTA displays any errors, refer to Chapter 8, Diagnostics.

6.16 Automatic Mode Operation

To run the sample program and check Automatic Mode operation:

1. Turn the selector switch to Automatic Mode (wired to connector X17, Pin 2).
2. Press the Cycle Start pushbutton (connected to X17, Pin 4)

Note: The DLC will execute the sample program and stop after one cycle. Change block 0004 to JMP 0000 to run the cycle continuously.

3. Press the Cycle Stop pushbutton to stop operation. Press Cycle Start to restart the program.

Note: $\overline{\text{Cycle Stop}}$ will cause an immediate stop without error. $\overline{\text{Emergency Stop}}$ will cause an immediate stop also, but will cause a fault to be displayed. De-actuate the E-stop switch and press the CL (clear) key on the CTA keypad to clear the fault, to allow restarting the operation.

7 Serial Interface

The DLC Control Card includes a multi-format RS-232/485 port for two-way communication of programs, parameters and system status between the DLC and a host device. The interface protocol is designed to easily transmit and receive data to and from the DLC. This chapter describes the protocol and other communication requirements.

The host device must strictly adhere to the communications format as described in this chapter to achieve proper communication, or one of several "RS Format Errors" will be returned through the port (or indicated on the control status display, see Figure 2.2 for a map of displays) indicating that the information was not properly formatted or understood.

User programs (Block information) can be downloaded to the DLC when it is in any mode of operation (Auto, Manual, or Parameter). The same is true for system status. System parameters can be downloaded to the DLC, **only** when it is in Parameter Mode. If attempted in any other mode, the host receives a "Invalid Mode" error message through the port. Parameters can be read from the DLC in any mode.

The optional Indramat program assembler software (MotionManager™) allows you to write and edit user programs and parameter list on any DOS based computer. You can download these into the DLC control through the serial interface. You can also upload the information from the DLC and make a print out of your program and parameter files.

The optional Indramat SOT (Station Operator Terminal) is a remote mounted, operator control device for the DLC. It is used to communicate program and parameter information between the DLC and SOT. The software in the SOT includes Help screens to assist the operator in using the SOT and for entering information correctly. The Indramat command line editor (Screen Manager), is used to write information and prompt lines for the operator that will appear on the SOT display (16 lines, 40 characters each). Contact Indramat for additional information on these options.

7.1 Connector Wiring (DB-9)

The serial interface connector (standard DB 9), is the X31 connector, located on the front of the DLC Control Card.

Refer to Figure 7.1 for a pin-out diagram of the multi-function port (X31 connector). It shows the pin numbers and definitions of signal connections for each type of communication.

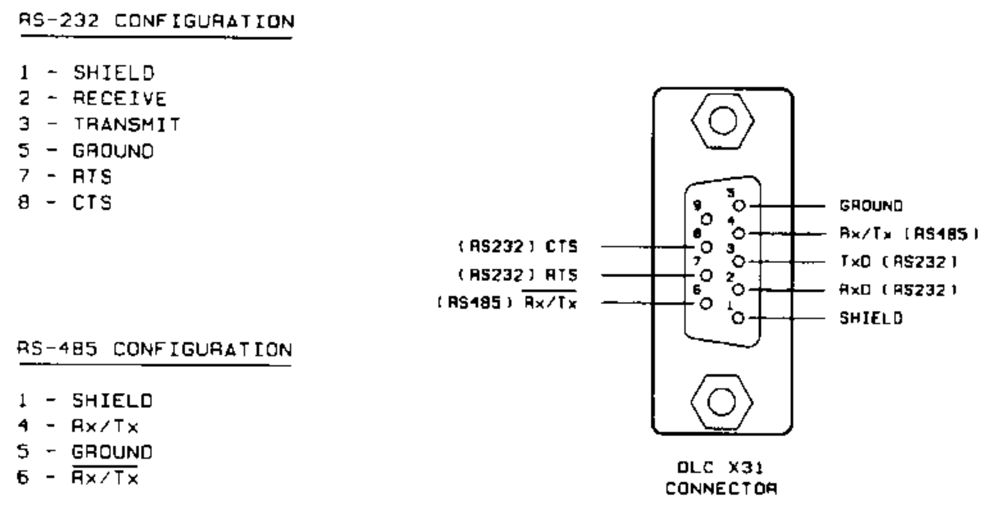


Fig. 7-1: DLC X31 Connector

Signal Level Requirements

Figure 7.2 illustrates the signal level requirements for the different communications (RS232/485) for the DLC Control Card. To minimize signal degradation over long cable runs, the serial device driver should provide the following levels:

- RS-232 ± 15 Vdc (50 ft maximum run)
- RS-422 ± 5 Vdc (1000 ft maximum run)
- RS-485 ± 5 Vdc (3000 ft maximum run)

The length of the transition for each word "T" is set in the serial interface parameter B003. The current hardware version is capable of 300 to 19200 baud.

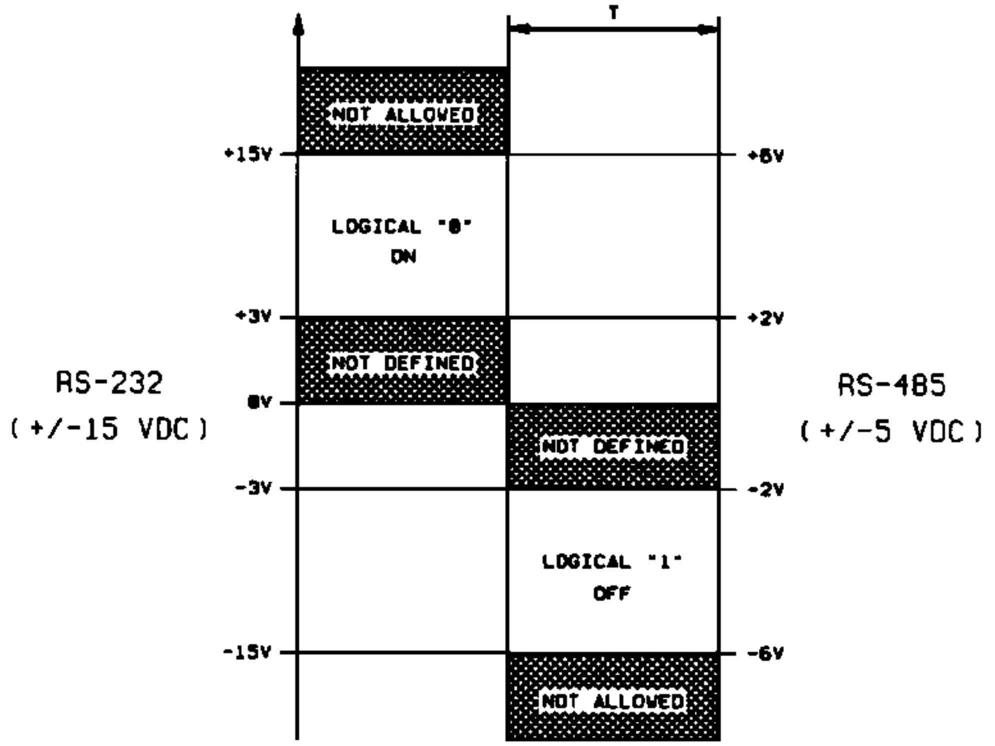


Fig. 7-2: Signal Level Requirements

Serial Cable Configurations

For RS-232 serial communications, you should note that the connector on the serial card in your computer can vary in configuration. Figure 7.3 illustrates two common serial connections for interfacing from the RS-232 port of a computer to the DLC.

```
DLC                COMPUTER              DLC                COMPUTER
9 pin D            25 pin D              9 pin D            9 pin D

TD 3 >-------------------> 3             TD 3> -------------------> 2
RD 2< -------------------< 2             RD 2< -------------------< 3
Grnd 5< -----------------> 7             Grnd 5<-----------------> 5

7 <------                                7 < -----
        |          9 pin to 25 pin               |          9 pin to 9 pin
8 <                D connector           8 <                D connector
```

Fig. 7-3: RS-232 Serial Cable Configurations

The DLC requires only three lines of the standard 9-pin connector for RS-232 communication. Pin 3 is for Transmitted Data, pin 2 for Received Data, and pin 5 is the Data Signal Ground. Typically, pin 1 connects the cable shield to ground on one end only. The transmit data connection from one device connects to the receive data of the other device, and vice versa. Signal ground connection must be common on both devices. Pins 7 and 8 are jumped on the DLC side only. Consult the manufacturers information for the serial card in your computer for its specific pin configuration. Indramat can supply a standard serial cable (cable # 05-0046) to connect from 9 pin PC to 9 pin DLC card. You may prefer to buy the required connectors and wire the cable in-house. Note that most serial cards do not provide a signal strong enough to go through a cable longer than about 50 feet. When possible, use a shielded cable no more than 25-50 feet long. **Only connect the pins shown in this figure. Wires connected on any pins other than the ones shown may cause communications errors.**

7.2 Data Format

To achieve proper communications, configure the communication parameters (B003 and B004) to match between your computer and the DLC. Figure 7.4 illustrates the data format. Following sections describe each parameter for data communication. Section 7.5 further describes the Checksum options of parameter B004.

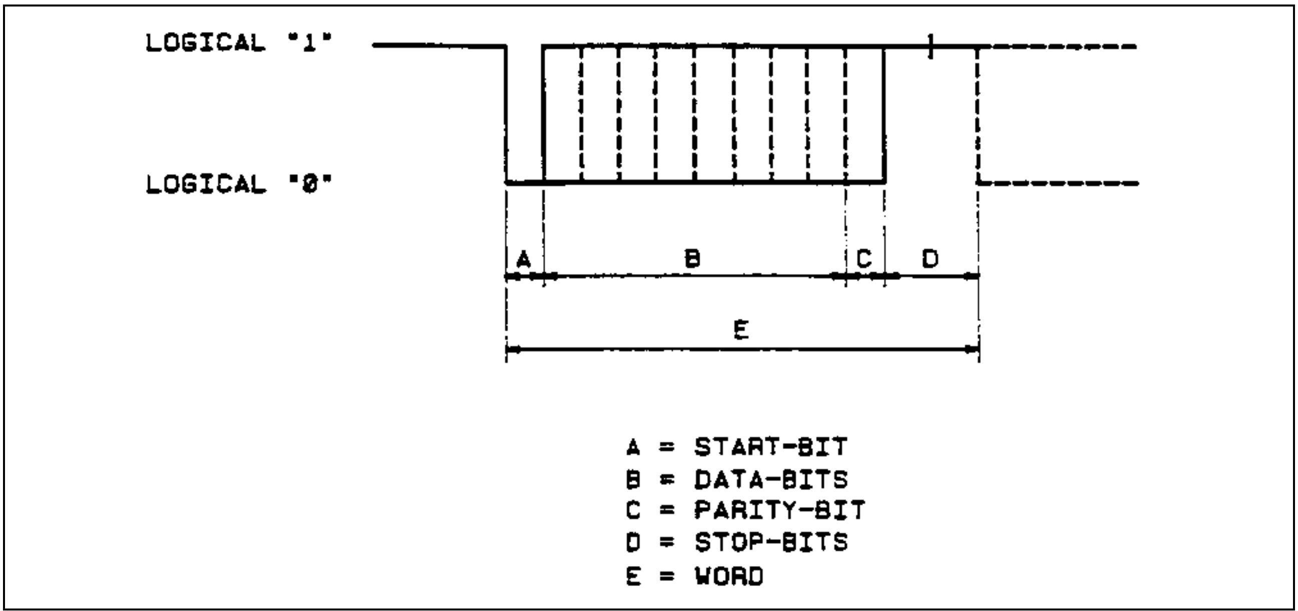


Fig. 7-4: Data Format

Word Length

The word length is set in parameter B003 (see Chapter 4 for entry procedures).

- 7 = 7 bit word length
- 8 = 8 bit word length

Parity Check

The parity type is set in parameter B003 (see Chapter 4 for entry procedures).

- 1 = no parity
- 2 = even parity
- 3 = odd parity

Baud Rate

The baud rate is set in parameter B003 (see Chapter 4 for entry procedures).

You may choose any baud rate from among the following values:

Parameter Entry	Baud Rate
0030	0300
0060	0600
0120	1200
0240	2400
0320	3200
0480	4800
0640	6400
0960	9600
1920	19200

Interface Mode

The interface mode is set in parameter B003 (see Chapter 4 for entry procedures).

You may choose from among the following:

0	Standard RS232 (Full Duplex)
1	IDS, decade switch option
2	Same as Mode 0
3	Serial port for SOT (Station Operator Terminal); RS-232 half duplex, one station ONLY
4	Serial bus for SOT; RS-485, half duplex, station 1 through 32 entered in parameter B004

7.3 DLC Control String Protocol

The following sections describe each control character requirements for proper protocol.

First (1) Control String Character (Transmission Type)

All data transmissions to the DLC must start with one of the following control characters to identify what type of transmission is to follow:

? Hexadecimal **3F**
The DLC interprets this character (received via the RxD channel) as a "Request for Information" - when followed by the proper requesting codes, the DLC will transmit the desired data via the TxD channel.

Hexadecimal **23**
This character signifies a block of "Information to be Stored" into memory. The data that follows will be read into the proper memory location.

! Hexadecimal **21**
This character signifies a "System Parameter" or "Control Command" is to follow.

Second (2) Control String Character (DLC Unit # Identifier)

This character is only present if communicating in the RS-485 mode. It is used to identify the DLC unit # to receive the current message. If communicating in RS-232 mode, this character will be a space (Hex 20).

Space		RS-232 Mode			
1	DLC#1	RS-485 Mode	H	DLC#17	RS-485 Mode
2	DLC#2	RS-485 Mode	I	DLC#18	RS-485 Mode
3	DLC#3	RS-485 Mode	J	DLC#19	RS-485 Mode
4	DLC#4	RS-485 Mode	K	DLC#20	RS-485 Mode
5	DLC#5	RS-485 Mode	L	DLC#21	RS-485 Mode
6	DLC#6	RS-485 Mode	M	DLC#22	RS-485 Mode
7	DLC#7	RS-485 Mode	N	DLC#23	RS-485 Mode
8	DLC#8	RS-485 Mode	O	DLC#24	RS-485 Mode
9	DLC#9	RS-485 Mode	P	DLC#25	RS-485 Mode
A	DLC#10	RS-485 Mode	Q	DLC#26	RS-485 Mode
B	DLC#11	RS-485 Mode	R	DLC#27	RS-485 Mode
C	DLC#12	RS-485 Mode	S	DLC#28	RS-485 Mode
D	DLC#13	RS-485 Mode	T	DLC#29	RS-485 Mode
E	DLC#14	RS-485 Mode	U	DLC#30	RS-485 Mode
F	DLC#15	RS-485 Mode	V	DLC#31	RS-485 Mode
G	DLC#16	RS-485 Mode	W	DLC#32	RS-485 Mode

Third (3) Control String Character (Information Type)

This character is used to identify the type of information to be sent.

N Hexadecimal 4E
Identifier for a program block. The information which follows this character will be stored as a program block (0000-2999).

K Hexadecimal 4B
Identifier for a system parameter. The information which follows this character will be stored as a parameter.

X Hexadecimal 58
Identifier for DLC status. The status type requested will be sent back to the host device.

Other Important Control Characters

The following are additional control characters required for proper protocol.

$ Hexadecimal **24**
Identifier for check sum. The two characters following this character represent the check sum of the information transmitted. This check sum must be transmitted along with every transmission.

CR Hexadecimal **OD**

LF Hexadecimal **OA**

These two characters, CR (Carriage Return) and LF (Line Feed) form the end of every transmission.

X-ON Hexadecimal **11**

X-OFF Hexadecimal **13**

Serial transmission can be controlled using handshaking.

- If the DLC is sending data via the TxD channel and receives the "X-OFF" signal (Hexadecimal 13/ASCII DC3) via the RxD channel, the DLC will interrupt the transmission until the "X-ON" signal (Hexadecimal 11/ASCII DC1) is received again via the RxD channel.
- If the DLC is receiving data via the RxD channel, and an interruption of the data transmission becomes necessary, the DLC will send the "X-OFF" signal (Hexadecimal 13/ASCII DC3) via the TxD channel. When the transmission can be resumed, the DLC will send the "X-ON" signal (Hexadecimal 11/ASCII DC1) via the TxD channel.

7.4 Information Characters

All information characters are coded in hexadecimal. The following characters are used for exchange of information:

0 through 9	Hexadecimal 30 through Hexadecimal 39
A through Z	Hexadecimal 41 through Hexadecimal 5A

Used as command codes for the DLC, depending on programming,

A - Z must be uppercase.

_ (space) Hexadecimal **20**

For creating the desired format, the space-character is used.

Note: This chapter indicates a space with the underline character.

+ Hexadecimal **2B**

- Hexadecimal **2D**

The operational sign must be transmitted for position command blocks

. Hexadecimal **2E**

, Hexadecimal **2C**

Used in data fields, depends on Language selected if responds with period (.) or comma (,). The DLC treats both the same.

7.5 CHECKSUM Calculations

If checksum is enabled in parameter B004, the following example shows how it would be calculated. After all characters are added together, the High-byte is added to the Low-byte, then the compliment of the two is taken. This number should immediately follow the "$" character.

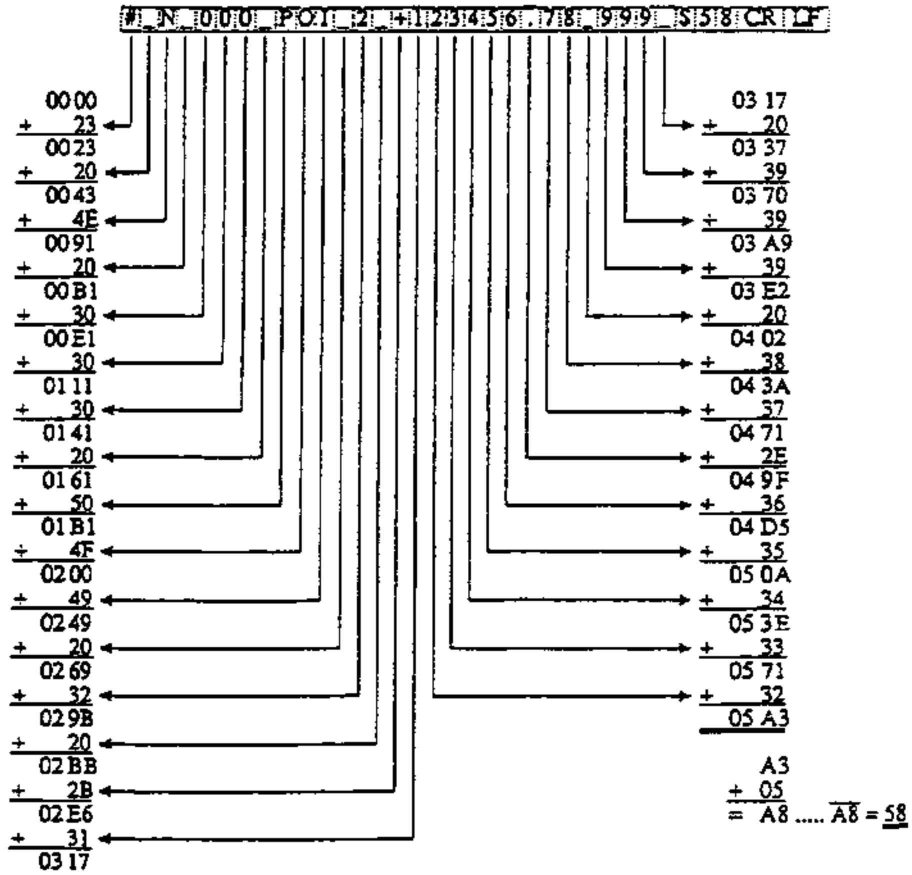


Fig. 7-5: CHECKSUM Example Calculation

7.6 Sending Information to the DLC

The DLC is capable of receiving new program block and parameter data from a host device, using the set of protocols described in the following sections. They use the key characters defined in sections 7.3 to 7.5, and a set data field to transfer the required information.

To get an acceptable transmission, it is very important that all characters, including all spaces, are used in the exact described format when sending to the DLC. If there is any type of discrepancy, the DLC will respond with an error message describing the type of format error that was found, and the current data will not be changed.

Sending Program Blocks to the DLC

A program block sent to the DLC must follow this format:

```
#_Nbbbb_ccc_dddddddddddddddd_$hhCRLF
```

The following describes each part of the command string:

#_N	Send program block to DLC
bbbb	Block Number (0000-2999)
ccc	Command Mnemonic
dd-->dd	Data, 16 characters in the proper format for a given command
$	End of block (check sum may follow)
hh	Check sum (if enabled in parameter 42)
CrLf	Carriage Return, Line Feed

Note: All transmitted data fields must be comprised of 16 characters. Send trailing spaces to fill the data block.

The following provides example of command strings for program blocks with different size data fields. Note the information as it appears on the DLC display, then the format required to send the information, filling all 16 characters positions of the data field.

Display Screens

```
E   0100 AEA
07   0
```

```
E   0101 PSI
1 +02345.678 999
```

Serial Data To Transmit

(same information that can be entered from the CTA keypad on the Edit display screens above)

```
             |                |
#_N0100_AEA_|07__0___________|_$5BCRLF
             |                |
             |   Data Field   |
             |****************|
             |                |
#_N0101_PSI_|1_+12345.678_999|_$53CRLF
             |                |
```

Refer to Chapter 5 for description of each command and its data field requirements.

Table 7.1 illustrates the serial data string arrangement for each command and its data field.

Sending Parameters to the DLC

Parameters sent to the DLC must follow this format:

```
! _ K _ y y x x _ d d d d d d d d _ $ h h CR LF
```

The following describes each part of the command string:

!_K	Send parameter to DLC
yy	Parameter Set
xx	Parameter Number
dd-->dd	Data, 8 characters in the proper format for a given parameter
$	End of block (check sum may follow)
hh	Check sum (if enabled in parameter B004)
CrLf	Carriage Return, Line Feed

Parameter Set	Code (yy=)	Parameter Number (xx=)
General parameter	B0	00 to 15
Axis 1 parameter	A1	00 to 25
Drive Parameter	C0	00 to 12
M-Function	MH or ML	00 to 63
Tool Corrections	DO	01 to 19

Note: The DLC must be in Parameter Mode before sending parameter information or an error will be issued.

7.7 Alphabetical Listing of Commands

ACC _ 1 _ _ 9 9 9 _ _ _ _ _ _ _ _ _ _ _	Acceleration Change
AEA _ 0 7 _ _ 1 _ _ _ _ _ _ _ _ _ _ _ _	Auxiliary Output ON/OFF
AKN _ 0 7 _ _ 1 _ _ _ _ _ _ _ _ _ _ _ _	Acknowledge Single Input
AKP _ 0 _ _ _ _ _ 0 0 0 1 1 1 2 2 2 0	Parallel Acknowledgment Input
APE _ 0 _ _ _ _ _ 0 0 0 1 1 1 2 2 2 0	Parallel Outputs ON/OFF
APJ _ 1 2 3 _ 4 _ 0 0 0 1 1 1 2 2 2 0	Set Parallel Outputs, then Jump
ATS _ 0 7 _ _ 1 _ _ _ _ _ _ _ _ _ _ _ _	Output State Monitor
BAC _ 3 4 5 _ _ + 1 2 3 4 _ 1 2 3 4 5	Branch And Count
BCA _ 3 4 5 _ _ 0 7 _ _ 1 _ _ _ _ _ _	Output-Dependent Conditional Branch
BCB _ 3 4 5 _ 2 0 _ _ 1 _ _ _ _ _ _ _	Binary Input Conditional Branch
BCD _ 3 4 5 _ 2 0 _ _ _ _ _ _ _ _ _ _ _	BCD-Dependent Conditional Branch
BCE _ 3 4 5 _ _ 0 7 _ _ 1 _ _ _ _ _ _	Input-Dependent Conditional Branch
BIC _ 1 2 3 4 _ 2 0 _ _ 1 5 _ 3 _ 0 _	Branch Input/Output Compare
BIO _ 3 4 5 _ 1 _ 0 0 0 1 1 1 2 2 2 0	Branch Input/Output Compare
BMB _ 1 0 0 _ 1 0 _ 0 4 _ 8 _ _ _ _ _	Binary Output-Dependent Conditional Branch
BPA _ 3 4 5 _ 1 _ 0 0 0 1 1 1 2 2 2 0	Branch on Parallel Outputs
BPE _ 3 4 5 _ 1 _ 0 0 0 1 1 1 2 2 2 0	Branch on Parallel Inputs
BPT _ 3 4 5 _ 2 _ + 1 2 3 4 5 . 6 7 8	Branch If Position Has Been Reached
BZP _ 3 4 5 _ 2 _ + 1 2 3 4 5 . 6 7 8	Branch if Target Position Exceeds Position Limit
CID _ 3 4 5 _ 1 _ 0 _ + 1 2 3 4 5 6 7	Change Instruction Data
CIO _ 0 _ 0 1 _ 2 3 _ 4 _ _ _ _ _ _ _	Copy Input/Output to Output
CLA _ 1 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Clear axis (Absolute Encoder Value)
CLC _ 3 4 5 _ _ _ _ _ _ _ _ _ _ _ _ _	Clear Counter
COC _ 1 _ 0 7 _ 0 0 1 1 2 2 _ + 3 6 0	Cam Output Control
CON _ 1 _ _ 0 _ + 9 9 9 _ 0 7 _ _ _ _	Continuous Operation (ON/OFF)
COU _ + 1 2 3 4 5 _ 1 2 _ 1 2 3 4 5 6	Count
CPL _ 1 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Clear Position Lag
CST _ 1 _ 1 _ _ _ _ _ _ _ _ _ _ _ _ _	Change Subroutine Stack - Pointer
D== _ 1 2 _ _ _ _ _ _ _ _ _ _ _ _ _ _	Tool Correction Memory Selection
FAK _ 1 _ _ 1 . 2 3 4 5 6 7 _ _ _ _ _	Factor All Motions (All Positions by X)
FOL _ 1 _ _ 2 _ _ 1 . 2 3 4 5 6 7 _ _	axis Synchronization (on/off)
FUN _ _ 2 _ 2 _ 1 _ 2 2 2 2 _ _ _ _ _	Functions

Rexroth Indramat

F== _ 1 2 3 4 5 6 7 8 _ _ _ _ _ _ _ _	Feedrate command for G-Code Functions
G01 _ 1 _ + 1 2 3 4 5 6 7 8 _ 9 9 _ _	Linear Feed with M Functions
G04 _ 1 2 3 4 _ _ _ _ _ _ _ _ _ _ _ _	Dwell Time
G40 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Deactivate Tool Correction
G43 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Add Tool Correction to Command Position
G44 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Substract Tool Correction to Command Position
G60 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Position with Exact Stop
G61 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Begin Velocity Rate Optimization Profile
G74 _ 1 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Reference (Home) Axis
G90 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Begin Absolute Positioning
G91 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Begin Incremental Positioning
HOM _ 1 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Home axis
JMP _ 3 4 5 _ _ _ _ _ _ _ _ _ _ _ _ _	Unconditional Jump (to block)
JSR _ 3 4 5 _ _ _ _ _ _ _ _ _ _ _ _ _	Jump to Subroutine
JST _ 3 4 5 _ _ _ _ _ _ _ _ _ _ _ _ _	Jump and Stop
JTK _ 3 4 5 _ 1 _ _ _ _ _ _ _ _ _ _ _	Jump in Task (task interrupt)
KDI _ 2 0 0 _ _ 1 0 0 _ 1 _ _ _ _ _ _	Copy Position Difference
MOM _ 1 _ 1 2 3 _ 1 2 3 _ 1 2 _ 1 2 3	Torque Reduction
M== _ 1 2 _ _ _ _ _ _ _ _ _ _ _ _ _ _	Select M-Function
NOP _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Blank Block (no operation)
PBK _ 1 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Position Break
PFA _ 1 _ + 1 2 3 4 5 . 6 7 8 _ 9 9 9	Absolute Position Feed to Positive Stop
PFI _ 1 _ + 1 2 3 4 5 . 6 7 8 _ 9 9 9	Incremental Position Feed to Positive Stop
POA _ 1 _ + 1 2 3 4 5 . 6 7 8 _ 9 9 9	Position Absolute Feed
POI _ 1 _ + 1 2 3 4 5 . 6 7 8 _ 9 9 9	Position Incremental Feed
POM _ 1 _ 0 _ _ _ _ _ _ _ _ _ _ _ _ _	Incremental Feed to Decade Switch Position
PSA _ 1 _ + 1 2 3 4 5 . 6 7 8 _ 9 9 9	Absolute Feed with Position Acknowledgment
PSI _ 1 _ + 1 2 3 4 5 . 6 7 8 _ 9 9 9	Incremental Feed with Position Acknowledge
PSM _ 1 _ 0 _ _ _ _ _ _ _ _ _ _ _ _ _	Feed to Decade Switch Pos. w/ Acknowledge
PST _ 1 _ _ 0 1 _ + 1 2 3 4 5 . 6 7 8	Position Test
REF _ 1 _ 0 _ 9 9 9 _ 1 2 _ _ _ _ _ _	Detect Registration Mark Input
REP _ 3 4 5 _ _ _ _ _ _ _ 1 2 3 4 5 6	Registration Search Limit Branch
RMI _ _ 0 _ 0 1 _ _ _ _ _ _ _ _ _ _ _	Registration Mark Interrupt
RSV _ 1 _ 0 0 0 _ 1 0 0 0 0 0 _ _ _ _	Restart Vector
RTS _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	Return from Subroutine
SAC _ 1 _ 0 _ _ _ + 1 2 3 4 5 . 6 7 8	Set Absolute Counter
SIN _ 1 _ _ 0 7 _ 1 2 . 3 4 5 _ 1 2 3	Sine Oscillation
SO1 _ 1 _ 1 _ 0 7 _ 9 9 9 _ _ _ _ _ _	Scanning of Inputs and Modifying a Length
STH _ 0 _ _ 0 0 1 _ _ _ _ _ _ _ _ _ _	Send to Host
STO _ 1 2 3 _ 1 _ 2 _ 1 2 3 4 _ 1 2 _	Send Information to Outputs
VCA _ 1 _ + 1 2 3 4 5 . 6 7 8 _ 9 9 9	Velocity Change Absolute
VCC _ 1 _ _ 1 2 3 4 5 . 6 7 8 _ 9 9 9	Velocity Change
VEO _ 1 _ 1 _ 1 _ 9 9 9 _ 0 _ _ _ _ _	Velocity Override Command
WAI _ 0 1 . 0 0 _ _ _ _ _ _ _ _ _ _ _	Time Delay (Wait)
WRI _ 3 4 5 _ _ 0 1 1 _ _ _ _ _ _ _ _	Write in Absolute Position (teach command)

7.8 Information Request

Program, Parameter and System Status can be requested from the DLC. This allows a host device to discern the status of the DLC and access the following information:

- Request Program Block
- Request Parameter
- Request Status Information

Each is fully described in the following sections.

Requesting a Program Block from the DLC

A request for a program block begins with "?_N" and ends with "CrLf" - a checksum is not required. Request format is as follows:

```
? _ N b b b b _ CR LF
```

The following describes each part of the command string:

?_N	Send program block to host
bbbb	Block Number (0000-2999)
CrLf	Carriage Return, Line Feed

The DLC will send the requested program block in the following format (refer to Table 7.1 for illustration of the data format for each command):

```
# _ N b b b b _ c c c _ d d d d d d d d d d d d d d d d _ $ h h CR
LF
```

The following describes each part of the command string:

#_N	Sending program block to host
bbbb	Block Number
ccc	Command Mnemonic
dd-->dd	Program block Data
$	End of block (check sum may follow)
hh	Check sum (if enabled in parameter 42)
CrLf	Carriage Return, Line Feed

Requesting a Parameter from the DLC

It is not necessary to be in Parameter Mode to request parameter data.

The computer Sends a Request in the format as follows:

```
? _ K _ y y x x _ CR LF
```

The following describes each part of the command string:

?_K	Send parameter to host
yy	Parameter Set
xx	Parameter Number
CrLf	Carriage Return, Line Feed

The DLC responds to the requested parameter in the following format:

```
! _ K _ y y x x _ d d d d d d d d _ $ h h CR LF
```

The following describes each part of the command string:

!_K	Sending parameter to host
yy	Parameter Set
xx	Parameter Number
dd-->dd	Parameter Data
$	End of block (check sum may follow)
hh	Check sum (if enabled in parameter B004)
CrLf	Carriage Return, Line Feed

Parameter Set	**Code (yy=)**	**Parameter Number (xx=)**
General parameter	B0	00 to 15
Axis 1 parameter	A1	00 to 25
Drive parameter	C0	00 to 12
Tool correction value	DO	01 to 19
M-Function	MH or ML	00 to 63

Examples: Query: ?_K_B003 (Serial Interface)
Reply: !_K_B003_19200181

Query: ?_K_A108 (Feed Constant axis 1)
Reply: !_K_A108_00100000

Requesting System Status from the DLC

The following System Status Information can be requested from the DLC.

00	Axis 1 Current Position (Decimal)
01	Transmission Error # And Text
02	Current Program Block Operating (Task 1 Only)
03	Axis 1 Current Position (Hexidecimal)
04	Counter status
05	Software version
06	Input status
07	Output status
08	Current Program Block And Subroutine Operating(All Tasks)
09	Current Axis 1 & Measuring Wheel Encoder Position
10	Axis 1 Position Lag
14	Commanded Current at Position
18	Drive Diagnostics
19	Hardware and Software Version
46	Length Counter
47	Actual RPM, Measuring Wheel
48	Axis 1 Current Motor RPM
50	System inputs and outputs (Hex)
51	User inputs (1-88) (Hex)
52	User outputs (1-96) (Hex)
53	System Faults

Each Status Request and DLC Response is described on the following pages.

Status 00 = Axis 1 Current Position (Decimal)

Request Format:

```
? _ X _ _ 0 0 _ Cr Lf
```

Response Format:

```
X _ 0 0 _ e d 1 1 1 1 1 . 1 1 1 _ x x x x x x x . x x x_ $ h h Cr
Lf
```

e =	"_" if axis has not been Homed, "A" if axis has been Homed
d =	Direction (+/-)
11111.111 =	Current position of axis 1 (in decimal format, in input units)
xxxxx.xxx =	Not Used

Status 01 = RS Transmission Error # and Text

This information is sent automatically by the DLC when there is a RS communication format error received via the port.

This information cannot be requested.

Transmission Format:

```
X _ 0 1 _ e e _ t t t t t t t t t t t t t t t t _ _ _ _ _ $ h h Cr
Lf
```

e =	Error Number
t =	Error Text

Consult Chapter 8, Diagnostics and Troubleshooting, for more information on the serial communication error used in the DLC Control Card.

Status 02 = Current Program Block (for Task 1 only)

Request Format:

```
? _ X _ _ 0 2 _ Cr Lf
```

Response Format:

```
X _ 0 2 _ N N N N _ n n n n _ $ h h Cr Lf
```

N =	Current Block Number
n =	Return to Main program Block number, if in a sub-routine (JSR Stack pointer)

Status 03 = Axis Current Position (Hexadecimal Format)

Request Format:

```
?_X_ _ 03 _ Cr :f
```

Response Format

```
x_03_z z z z z z z z___ y y y y y y y y _ _ _ _ _ _ _ $ h h Cr Lf
```

zzzzzzzz =	Current position of axis 1 (in Hexadecimal format)
yyyyyyyy =	Not Used

Response is in Hexadecimal. Use the following formulas to convert to Input Units to axis 1 current position in decimal format.

For Linear: IU's = $\frac{zzzzzzzz * A108}{A104 \times 4}$

For Rotary: IU's = $\frac{zzzzzzzz * A108}{360° \ (A105)}$

Status 04 = Counter Status

Request Format:

```
? _ X _ _ 0 4 _ N N N N _ Cr Lf
```

Response Format:

```
X _ 0 4 _ N N N N _ a a a a a a _ t t t t t t _ _ _ _ _ _ $ h h Cr Lf
```

NNNN=	Counter block number
aaaaaa=	Actual count
tttttt=	Target count

Status 05 = Software Version

Request Format:

```
? _ X _ _ 0 5 _ Cr Lf
```

Response Format:

```
X _ 0 5 _ _ v v v v v v v v v v v v v v v v _ $ h h Cr Lf
```

v =	Software version (as displayed on the CTA display)

Example: "___DA 01.1-03.02___"

Status 06 = Input Status

Request Format:

```
? _ X _ _ 0 6 _ b _ Cr Lf
```

Response Format:

```
X _ 0 6 _ b _ e e e e e e e e e e e e e e e e _ $ h h Cr Lf
```

16 inputs are always represented

b=	Bank number (Input range)	
0=	System Inputs 1-8	(DEA 4.1 card)
1=	Aux. Inputs 1-7	(DEA 4.1 card)
2=	Aux. Inputs 08-22	(DEA 5.1 card)
3=	Aux. Inputs 23-37	(DEA 6.1 card)

e=	State of each of 15 inputs in sequence, as follows:
0=	OFF (0 Vdc)
1=	ON (24 Vdc)

Status 07 = Output Status

Request Format:

```
? _ X _ _ 0 7 _ b _ Cr Lf
```

Response Format:

```
X _ 0 7 _ b _ a a a a a a a a a a a a a a a a _ $ h h Cr Lf
```

16 outputs are always represented.

b =	**Bank number (Output range)**
0 =	System Outputs 1-5
1 =	Aux. Outputs 1-16
2 =	Aux. Outputs 17-32
3 =	Aux. Outputs 33-48
4 =	Aux. Outputs 49-64
5 =	Aux. Outputs 65-80
6 =	Aux. Outputs 81-96
7 =	Aux. Outputs 97-99

a =	**State of each of 16 outputs in sequence, as follows:**
0 =	OFF (0 Vdc)
1 =	ON (24 Vdc)

Status 08 = Current Block (for all Tasks)

Request Format:

```
? _ X _ _ 0 8 _ Cr Lf
```

Response Format:

```
X _ 0 8 _ 1 1 0 0 _ 1 2 0 0 _ 1 3 0 0 _ 1 4 0 0 _ 1 5 0 0 _ 1 6 0
0 _ $ h h Cr
                                  Lf
```

1100 =	Task 1 - Current block number
1200 =	Task 1 - Main program block number
1300 =	Task 2 - Current block number
1400 =	Task 2 - Main program block number
1500 =	Task 3 - Current block number
1600 =	Task 3 - Main program block number

The DLC returns the current block number of each task.

It also will return the main program block number if a given task is in a sub-routine (first level of the JSR stack pointer).

If not in a sub-routine, the current block number is repeated for the main program return block.

If a task has not been activated, spaces will be returned in the appropriate locations.

Status 09 = Measuring Wheel / Motor Encoder (Position Readings)

Request Format:

```
? _ X _ _ 0 9 _ Cr Lf
```

Response format:

```
X _ 0 9 _ _ x ± 1 2 3 4 5 . 6 7 8 _ ± 1 2 3 4 5 . 6 7 8 _ $ h h Cr
Lf
```

X_09 =	Status 09
_ =	Reserved - not used at present
x =	"A" - Axis 1 has been homed "_" - Axis 1 has not been homed
±12345.678 =	First Position Axis 1 in EGE - if Measuring Wheel activated, all MW movements are used - if MW deactivated, all motor transmitter movements are used
±12345.678 =	Second Position Axis 1 in EGE

Status 10 = Axis 1 Position Lag

Request Format:

```
? _ X _ _ 1 0 _ Cr Lf
```

Response Format:

```
X _ 1 0 _ d 1 1 1 1 1 . 1 1 1 _ x x x x x x . x x x _ $ h h Cr Lf
```

X_10 =	Status 10
d =	Direction (+/-)
11111.111 =	Position Lag of Axis 1 (in input units)
xxxxx.xxx =	Not Used

Status 14 = Command Current at Position

Request Format:

```
? _ X _ _ 1 4 _ Cr Lf
```

Response Format:

```
X _ 14e±12345.678±123.45$hh Cr Lf
```

X_14 =	Status 14
e =	"A" - Axis 1 has been homed "-" - Axis 1 has not been homed
±12345.678 =	Position of Axis 1
±123.45 =	Commanded current in amps

Status 18 = Drive Diagnostics

Request Format:

```
? _ X _ _ 1 8 _ Cr Lf
```

Response Format:

```
X _ 18 123 _ t t t t t t t t t t t t t t t t _ _ _ _ $ h h Cr Lf
```

X_18 =	Status 18
123 =	Drive diagnostic code
t =	Drive diagnostic text

Status 19 = Hardware and Software Version

Request Format:

```
? _ X _ _ 1 9 _ Cr Lf
```

Response Format:

```
X _ 1 9 _ H H H H H H H H H H H H H H  _ _ S S S S S S S S S S S S
S S _ _ $ h h Cr Lf
```

X_19 =	Status 19
H =	Hardware Version (example: __DLC-1.1-A__)
S =	Software Version (example: __DA01.1-03.02__)

Status 46 = Length Counter

Request Format:

```
? _ X _ _ 4 6 _ A _ Cr Lf
```

A = axis number (1)	0 = Measuring wheel

Note: When A = 0: If the measuring wheel option has not been set in parameters B 009 and B 010, the error message 'M-Wheel P.' will be sent instead of Status 46.

Response Format:

```
X _ 4 6 _ A _ d 1 2 3 4 5 . 6 7 8 _ $ h h Cr Lf
```

X_46 =	Status 46
A =	axis number (1), measuring wheel (0)
d =	Direction (+/-)
12345.678 =	Length selected axis has fed between the programmed ON and OFF of the command FUN (in input units)

Status 47 = Actual RPM, Measuring Wheel

Request Format:

```
? _ X _ _ 4 7 _ Cr Lf
```

Response Format:

```
X _ 4 7 _ O _ d 1 2 3 4 . 5 6 _ $ h h Cr Lf
```

X_47 =	Status 47
d =	Direction (+/-)
o =	Measuring Wheel axis
1234.56 =	Actual RPM of the Measuring Wheel

Note: If the measuring wheel option has not been set in parameters B 009 and B 010, the error message '10M-Wheel P. False' will be sent instead of Status 47.

Status 48 = Axis Current Motor RPM

Request Format:

```
? _ X _ _ 4 8 _ Cr Lf
```

Response Format:

```
X _ 4 8 _ 1 _ d 1 2 3 4 . 5 6 _ x _ y z z z z . z z _ $ h h Cr Lf
```

X_48 =	Status 48
1 =	axis number 1
d =	Direction (+/-)
1234.56 =	Actual RPM of axis 1
x =	Not Used
y =	Not Used
zzzz.zz =	Not Used

Status 50 = System Inputs and Outputs Hexadecimal

Request Format:

```
? _ x _ _ 5 0 _ Cr Lf
```

Response Format:

```
X _ 5 0 _ e e e e a a a a $ h h Cr Lf
```

e =	Four system inputs (Hexadecimal)
a =	Four system outputs (Hexadecimal)

.1 Hexadecimal Conversion Table

Weight =	**2^3**	**2^2**	**2^1**	**2^0**	**e/a**
	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	2
	0	0	1	1	3
	0	1	0	0	4
	0	1	0	1	5
	0	1	1	0	6
	0	1	1	1	7
	1	0	0	0	8
	1	0	0	1	9
	1	0	1	0	A
	1	0	1	1	B
	1	1	0	0	C
	1	1	0	1	D
	1	1	1	0	E

.2 Status 50 Example:

```
X _ 5 0 _ 0 0 7 4 0 0 F 9 & h h CR LF

0000      0000     0111      0100            0000      0000     1111     1001
16......  System Inputs ....  ........ 1     16.....   System Outputs ....  ......... 1
```

In this example:

- System inputs 3, 5, 6, and 7 are at +24 volts.
- System outputs 1, 4, 5, 6, 7, and 8 are at +24 volts.

Status 51 = User Inputs (Hexadecimal)

Request Format:

```
? _ X _ _ 5 1 _ CR  LF
```

Response Format:

```
X _ 5 1 _ e e e e e e e e e e e e e e e e e e e e $ H h CR LF
```

Input numbers (e = user input, hexadecimal code)

```
      e  e  e  e  e  e  e  e  e  e  e  e  e  e  e  e  e  e  e  e
      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

                                 36 32 28 24 20 16 12 8  4  =23
                                 35 31 27 23 19 15 11 7  3  =22
                                 34 30 26 22 18 14 10 6  2  =21
                              37 33 29 25 21 17 13  9 5  1  =20
0 is transmitted for all other inputs that do not exist in hardware.
```

Status 52 = User Outputs (Hexadecimal)

Request Format:

```
? _ X _ _ 5 2 _ CR  LF
```

Response Format:

```
X _ 5 2 _ a a a a a a a a a a a a a a a a a a a a a a a a a $ H h CR
LF
```

Output numbers (a = user output, hexadecimal code)

```
 a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  a a
 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

 96 92 88 84 80 76 72 68 64 60 56 52 48 44 40 36 32 28 24 20 16 12 8 4  =2^3
 95 91 87 83 79 75 71 67 63 59 55 51 47 43 39 35 31 27 23 19 15 11 7 3  =2^2
 94 90 86 82 78 74 70 66 62 58 54 50 46 42 38 34 30 26 22 18 14 10 6 2  =2^1
    93 89 85 81 77 73 69 65 61 57 53 49 45 41 37 33 29 25 21 17 13  9 5 1  =2^0
```

Status 53 = System Fault

Request Format:

```
? _ X _ _ 5 3 _ Cr Lf
```

Response Format:

```
X _ 5 3_ n n _ t t t t t t t t t t t t t t t t _ $ h h Cr Lf
```

n n =	error code (hexadecimal)
t =	Fault message text

Fault message texts and error codes are listed in detail in chapter 8.

8 Diagnostics And Troubleshooting

This chapter is divided into three sections. The first section, 8.1, describes the normal operating diagnostics of the digital drive. **If an error is displayed which is not described in Section 8.1, consult the Indramat Digital AC Servo Drive Manual for more information**.

Section 8.2 describes the system error codes and messages for the DLC Control Card and suggests possible solutions to the error conditions.

Section 8.3 describes the serial communications errors and suggests possible solutions to the error condition.

8.1 Digital AC Servo Drive Normal Operating Diagnostics

Error Message	Code #
Drive Enabled	102

Explanation of Diagnostic: The digital drive has powered up properly and the DLC Control card has enabled the servo system. The DLC system can now be operated in Automatic or Manual mode.

Solution: Not Required

Error Message	Code #
Drive Halt	103

Explanation of Diagnostic: The DLC Control card has detected an error. The digital drive will stop with the maximum torque available. The digital drive will then maintain zero velocity, with torque still applied to the motor, and remain in a closed position loop.

Solution: Check the DLC display to determine the DLC error detected. Then consult the "Solution" section under that specific diagnostic.

Error Message	Code #
Drive Ready	101

Explanation of Diagnostic: Control voltage and main power are present, (DC bus voltage is present). Drive is ready to be enabled. If the DLC Control Card has an error, the digital drive is typically displaying this diagnostic.

Check the DLC display to determine the error in the DLC. Then consult the "Solution" section under the diagnostic explanation.

Solution: Not Required

Error Message	Code #
Operation Ready	100

Explanation of Diagnostic: The digital drive is ready to have 3 phase power applied.

Solution: Not Required

Error Message	Code #
Safety Interlock	104

Explanation of Diagnostic: This diagnostic will occur when the AS+ input is applied. This input is on the Indramat Digital Drive, connector X3, pins 8 & 9. When the AS + input is applied, the digital drive will immediately command the motor to zero velocity and then disable the power section of the digital drive.

Solution: This message is a direct response to an external signal applied to the digital drive. Check machine wiring to determine why the AS input is being applied. Consult the manual for the specific digital drive for more information on the use of the AS inputs.

8.2 DLC System Error Code And System Error Messages

When the DLC detects an error, all outputs are turned off and the error message appears on the CTA display. After the error has been corrected, the message must be cleared by pressing the CL key on the CTA keypad, or by providing a +24-volt input to the external Clear input (connector X17, pin 8).

Typically, the word "ERROR" appears on the top line of the display. The second line displays one of the following messages:

Error Message	Code #
(No diagnostic message is displayed)	00

Explanation of Diagnostic: When the DLC Control card is operating properly , with no system errors present, requesting the Status Code 53 through an external host device causes the DLC Control card to transmit the error code "00".

No error message will be displayed on the CTA panel, if used. If another MMI device is used, it may display a message in response to this information.

Solution: Not Required

Error Message	Code #
Abs. Range	45

Explanation of Diagnostic: This error occurs if the maximum number of turns of the multi-turn absolute feedback has been exceeded.

Solution: Verify that the distance traveled does not exceed the maximum number of turns of the absolute encoder.

Error Message	Code #
Axis 1 Not Homed	48

Explanation of Diagnostic: This error occurs if the DLC program encounters an absolute position command for the axis, but that axis has not yet been homed.

Solution:

1. Refer to Chapter 3, Section 3.2.4 for details on Homing.
2. Verify that the axis homing parameters, A110 through A114, are correct. The axis can be homed in Manual or Automatic mode. Refer to Chapter 4 for more information on the Homing parameters.
3. Verify that the axis position command in the DLC program is correct. Refer to Chapter 5 for more information on positioning commands.

Error Message	Code #
Axis No. False	16

Explanation of Diagnostic: Incorrect axis selected in a feed command. Must be axis #1 only.

Solution: Check program block feed command axis designation.

Error Message	Code #
Battery Is Low	04

Explanation of Diagnostic: This condition occurs when the Lithium battery which retains the DLC memory (program, parameters, counter status, tool correction values, M-Function set-up, etc.) during power OFF, is below minimum voltage level.

A battery test is made at DLC power up. If the battery is low, the error message is displayed. This message will also re-appear every ten minutes as a reminder. During Normal operation in any of the three operating modes (manual, automatic or parameter), a battery test is made every four hours. If the battery is still low, this diagnostic will re-appear, but no fault will be issued.

Solution:

1. If displayed on initial power up, reset the DLC by pressing the CL key on the CTA panel or by bringing the External Reset input high.
2. Replace the battery within two weeks of the first appearance of this diagnostic. To replace the battery, turn off the power to the DLC (it gets its power from the digital drive), remove the DLC card from the U2 slot in the drive and replace the lithium battery.

Contact the Indramat Service Department at 800-860-1055 if you have any questions.

Error Message	Code #
BCD Input	0D

Explanation of Diagnostic: This error will occur if a BCD command is encountered in the DLC program and the auxiliary inputs are not in a BCD format.

Solution:

1. The BCD program command is programmed incorrectly. Refer to Chapter 5 for information on the use of the BCD command.
2. Verify that auxiliary inputs 1 through 8 are in a BCD format.

Error Message	Code #
Configuration	1A

Explanation of Diagnostic: This error will occur when the DLC Control card and the digital drive are incorrectly configured. The firmware installed in the DLC Control card does not have the capability to operate with the present configuration.

Solution: Write down the current configuration and DLC firmware installed and contact the Indramat Service Department for more information.

Error Message	Code #
Division By Zero	07

Explanation of Diagnostic: This is an internal DLC error.

Solution:

1. Check for a possible acceleration setting of zero, or an acceleration change (ACC program command) to zero.
2. Contact the Indramat Service Department.

Explanation of Diagnostic: This error occurs when there is an initialization conflict between the DLC Control card and the digital drive.

Solution: Write down the current configuration and DLC firmware installed and contact the Indramat Service Department for more information.

Error Message	Code #
DPR Error	1C

Explanation of Diagnostic: This error occurs when there is an initialization conflict between the DLC Control card and the digital drive. This is an error in passing information over the dual port RAM between the DLC Control card and the digital drive.

Solution: Write down the current configuration and DLC firmware installed and contact the Indramat Service Department for more information.

Error Message	Code #
Drive 1 Not Ready	40

Explanation of Diagnostic: The digital drive being used did not allow the main contactor to energize , allowing 3 phase power to be applied.

Solution:

1. An error has occurred in the Indramat digital drive. See the specific Indramat drive manual to determine the cause for the fault in the drive.
2. The digital drive does not have power applied. Verify the power to the digital drive is correctly wired and the proper inputs are applied to bring up main power.

Error Message	Code #
Drive Runaway A1	42

Explanation of Diagnostic: The DLC uses a mathematical model to calculate the "normal" following error expected. This diagnostic will occur when the actual following error exceeds the calculated following error by a percentage greater than that which is programmed in parameter A122.

Solution:

1. The motor physically moved without being commanded to move.
2. The Monitoring Window percentage, parameter A122, is too small. The typical setting is 10%.

Error Message	Code #
Emergency Stop	03

Explanation of Diagnostic: This error occurs when the Emergency Stop input to the DLC (DEA 4, X17, pin 3) goes low or is lost.

Solution:

1. The machine E-Stop button has been pushed.
2. The Emergency Stop circuit has been interrupted. Consult the machine wiring diagrams to determine the cause.
3. The 04-0123 I/O cable is not connected to the DEA 4 card.
4. +24 Vdc must be applied to DEA 4, X17, pins 36 & 37. Also, the reference (0 Vdc) must be applied to the DEA 4, X17, pins 34 & 35.

Error Message	Code #
Encoder Fault	4C

Explanation of Diagnostic: This error will occur when the percentage of deviation between the external encoder position and the axis position is exceeded. The percentage of deviation is defined in parameter A123.

Solution:

1. The External encoder Lines/Rev and Feed Constant parameters, B017 & B018, are incorrect. Verify that the external encoder parameters are correct, B016 through B018.
2. Verify that the axis Feed Constant , A108, is correct.
3. The percentage defined in the Follow Axis parameter, A123, is too small.
4. Verify that the External encoder direction parameter, B016, is correct.
5. Slippage between the external encoder and the material or coupling. Try a lower acceleration to reduce the slippage.
6. The external encoder is wired incorrectly.

Error Message	Code #
Excess Pos Lag 1	43

Explanation of Diagnostic: The DLC uses a mathematical model to calculate the "normal" following error expected. This diagnostic will occur when the actual following error is greater than the calculated following error by a percentage greater than that which is programmed in parameter A122.

Solution:

1. The axis is commanded to a position, but the axis servo system has an excessive position lag.
2. The digital drive does not have power applied. Verify the power to the drive is correctly wired and the proper inputs are applied to bring up main power. See machine drawings for power u sequence requirements.
3. The Accel Rate parameters, A102 or A117, values may be too large. The motor may be unable to accelerate at the rate specified.
4. The motor may not be able to turn due to a mechanical bind.

Error Message	Code #
Feed Angle Loss	41

Explanation of Diagnostic: This error occurs if the Feed Angle auxiliary input, designated in parameter A120 goes low while the axis is moving.

Refer to Chapter 4 for more information on the Feed Angle auxiliary input.

Solution:

1. Verify that the Feed Angle auxiliary input is properly connected.
2. If the Feed Angle is attached to a press that operates continuously, either the press speed must be slowed down or the part length being processed must be shorter.

Error Message	Code #
Home Switch Pos	4B

Explanation of Diagnostic: This error occurs if the marker pulse of the axis feedback is closer than 1/16 revolution to the homing cam switch.

Solution:

1. Refer to Chapter 3, Section 3.2.4 for details.
2. Move the axis home limit switch a distance equal to or greater than 1/3 of the axis feed constant, parameter A108.

Error Message	Code #
IDS Brake Error	12

Explanation of Diagnostic: This error will occur if the optional IDS board is enabled, but the DLC does not detect the connection of the IDS board.

The IDS board is enabled in parameter B003 (Serial Interface).

Solution:

1. The cable between the IDS board and the DLC connector, X31, is not installed properly.
2. The DLC serial communication port is defective.
3. The IDS board is defective.

Error Message	Code #
IDS Checksum Error	13

Explanation of Diagnostic: This error will occur if the optional IDS board is enabled, but the DLC does not detect the connection of the IDS board.

The IDS board is enabled in parameter B003 (Serial Interface).

Solution:

1. The cable between the IDS board and the DLC connector, X31, is not installed properly.
2. If the IDS board is connected, then the wiring and its layout to the decade switch should be checked. Maximum cable length is 15m.

Error Message	Code #
IDS Data Too Long	15

Explanation of Diagnostic: This error will occur if the optional IDS board is enabled, but the DLC does not detect the connection of the IDS board.

The IDS board is enabled in parameter B003 (Serial Interface).

Solution:

1. The cable between the IDS board and the DLC connector, X31, is not installed properly.
2. The DLC serial communication port is defective.
3. The IDS board is defective.

Error Message	Code #
IDS Format Error	14

Explanation of Diagnostic: This error will occur if the optional IDS board is enabled, but the DLC does not detect the connection of the IDS board.

The IDS board is enabled in parameter B003 (Serial Interface).

Solution:

1. The cable between the IDS board and the DLC connector, X31, is not installed properly.
2. If the IDS board is connected, then the wiring and its layout to the decade switch should be checked. Maximum cable length is 15m.

Error Message	Code #
Invalid Block #	0E

Explanation of Diagnostic: The program contains a combination of offset, jump distance and BCB/BCD commands that causes the DLC program to jump to a program block greater that 2999.

If Task 3 is being used, enter the program for Task 3 in the appropriate program block location before enabling Task 3 in parameter B006. You must do it this way, because once you enable Task 3 in parameter B006, it will begin to run as soon as you leave parameter mode.

Solution:

1. The BCB, BCD and BMB program commands can result in a jump or branch to a target block greater than 2999. This jump is caused by a combination of an offset, jump distance, or binary input. If any of these commands are used, refer to Chapter 5 to determine if the program command is used properly.
2. The branch or jump command contains an asterisk (*) in the target block. Refer to Chapter 5 to determine if the program command is used properly.

Error Message	Code #
Invalid Mode!	08

Explanation of Diagnostic: This occurs when both the Automatic and Parameter inputs are held high at the same time.

Solution:

1. If you remove the parameter input, you must reset the DLC to continue.
2. If you remove the Automatic input, the DLC will be in Parameter Mode, with no error resulting.

Error Message	Code #
Invalid Prg Command	0A

Explanation of Diagnostic: This error occurred because the DLC program encountered an invalid program command.

If Task 3 is being used, enter the program for Task 3 in the appropriate program block location before enabling Task 3 in parameter B006. You must do it this way, because once you enable Task 3 in parameter B006, it will begin to run as soon as you leave parameter mode.

Solution:

1. The program command contains asterisks (*). Refer to Chapter 5 to determine if the program command is being used properly.
2. The auxiliary input or output programmed in the DLC program is 0 (zero).
3. The REP command was encountered before the REF command was executed in the DLC user program.

Error Message	Code #
Is INVALID	02

Explanation of Diagnostic: This message appears when the value stored in a DLC parameter exceeds the minimum or maximum limits. This message appears on the first line of the display. The second line displays the number of the affected parameter. The fourth line will display the parameter name.

Solution: The parameter containing the invalid data can be displayed by switching back into parameter mode and pressing the CL key on the CTA keypad, or by bringing the External Clear input (DEA card, X17, pin 8) high (+24Vdc). Consult Chapter 4 of this manual to determine the minimum and maximum values for the parameter in which the error occurred."

Error Message	Code #
JSR Nesting	0B

Explanation of Diagnostic: This nesting error occurs if the nesting depth of the programmed subroutines is greater than 127.

Solution:
1. Change the DLC program so that the number of nested subroutines does not exceed 127.
2. Check the program and if necessary, program JSR without RTS.

Error Message	Code #
Max Travel Lmt 1	47

Explanation of Diagnostic: This maximum travel error occurs in Automatic mode if the value stored in the Maximum Travel Limit parameter is exceeded. The axis commanded position has exceeded the maximum software travel limit.

Solution: 1) Verify that the axis maximum travel limit parameter is correct.

Error Message	Code #
Min Travel Lmt 1	46

Explanation of Diagnostic: This minimum travel error occurs in Automatic mode if the value stored in the Minimum Travel Limit parameter is exceeded. The axis commanded position has exceeded the minimum software travel limit.

Solution: Verify that the axis minimum travel limit parameter is correct.

Error Message	Code #
Parameter Mode	01

Explanation of Diagnostic: When the DLC Control card is put into Parameter mode, a host control request for Status 53 causes the DLC Control card to transmit this code.

Solution: Not Required

Error Message	Code #
Parameters Lost	05

Explanation of Diagnostic: When power to the DLC is turned off and the battery does not have sufficient power to retain its memory or is disconnected, this error occurs when power is re-applied. It signifies that the parameters that were stored in memory are no longer in memory.

This error may also occur it the firmware is removed or if a different DLC firmware is installed.

Solution:
1. If this error occurs every time the DLC is powered up, replace the backup battery.
2. Verify that every DLC parameter has valid data in it. The DLC parameters might contain asterisks (*), indicating invalid data has been entered. Refer to Chapter 4 for information on how to enter the DLC parameters.

Error Message	Code #
Program Lost	06

Explanation of Diagnostic: When power to the DLC is turned off and the battery does not have sufficient power to retain its memory or is disconnected, this error occurs when power is re-applied. It signifies that the program that was stored in memory is no longer in memory.

This error may also occur if the firmware is removed or if different DLC firmware is installed.

This error will also occur if you change the number of decimal places selected in parameter B007.

Solution:

1. If this error occurs every time the DLC is powered up, replace the backup battery.
2. Verify that every DLC program block has valid data in it. The DLC program might contain asterisks (*), indicating invalid data has been entered. Refer to Chapter 5 for information on how to enter the DLC program.
3. If you changed the number of decimal places in B007, pull up the Program Edit screen and press the block store key 2 or 3 times to store program blocks with the new decimal place settings. Press the reset key to clear the error.

Error Message	Code #
RTS Nesting	0C

Explanation of Diagnostic: This nexting error occurs when an RTS command is encountered in the DLC program without a matching JSR command.

Solution:

1. Verify in the DLC program that a JSR command appears before the RST command is encountered.
2. Refer to Chapter 5 for information on the use of JSR and RTS commands.

Error Message	Code #
Software Combination	18

Explanation of Diagnostic: This error will occur if the drive software in the DSM module is incompatible with the DLC software.

Solution: Contact Indramat Service Department.

Error Message	Code #
System Fault, “IRQ0VL Interrupt	11

Explanation of Diagnostic: This error is an internal software fault in the DLC.

Solution: The DLC card could not process all requested data within the 1 millisecond time slice allowed. Contact the Indramat Service Department.

Error Message	Code #
System Fault “Stack Overflow”	10

Explanation of Diagnostic: This error is an internal software fault in the DLC.

Solution: Contact the Indramat Service Department.

Error Message	Code #
TC Memory Lost	

Explanation of Diagnostic: When power to the DLC is turned off and the battery does not have sufficient power to retain its memory or is disconnected, this error occurs when power is re-applied. It signifies that the tool correction register data that was stored in memory is no longer in memory.

This error may also occur if the firmware is removed or if different DLC firmware is installed.

Solution:

1. If this error occurs every time the DLC is powered up, replace the backup battery.
2. Verify that every DLC tool correction register has valid data in it. The DLC tool correction registers might contain asterisks (*), indicating invalid data has been entered. Refer to Chapter 5 for information on how to enter the DLC tool correction values.

Error Message	Code #
WRITE Command	0F

Explanation of Diagnostic: The WRI command tried to write into a target block which did not contain a POA or PSA command.

Solution: Check the DLC program and refer to Chapter 5 to determine if the WRI command is used properly.

8.3 Serial Communication Errors

Serial communications errors are not displayed on the CTA display, but the serial communication error code and error message will be transmitted from the CLM to a host device (SOT, computer, etc.).

The correct format must be used when information is transmitted to the DLC from a host device. If the format is not correct, the DLC will transmit an error code and message from the DLC serial communications port (Connector X31) to the host device. The error code and message are transmitted with the status number 01. Refer to the Serial Interface, Chapter 7 for information on formats and how status 01 operates.

The DLC serial communication errors are as follows:

Error Message	Code #
Block # Too Large	01 12

Explanation of Diagnostic: The program block number transmitted to the DLC was too large.

Solution:
1. Verify that the block number is between 0000 and 2999.
2. Refer to Chapter 7, Section 7.6.1, for the correct format.

Error Message	Code #
Invalid Mode!	01 05

Explanation of Diagnostic: An attempt was made to transmit parameter information without first selecting Parameter mode.

Solution: Verify that Parameter mode is selected before transmitting parameter information to the DLC.

Error Message	Code #
Invalid P.-Block	01 11

Explanation of Diagnostic: The parameter format transmitted to the DLC was incorrect.

Solution:
1. Verify that the parameter format is transmitted correctly to the DLC.
2. Refer to Chapter 7, Section 7.6.2, for the correct format.

Error Message	Code #
Invalid Prg. Command	01 13

Explanation of Diagnostic: The program command being transmitted to the DLC is invalid. It is not included in the list of programming commands found in Chapter 5.

Solution:
1. Verify that the program command being transmitted to the DLC is listed in Chapter 5.
2. Verify that the command was entered correctly. See Chapter 7, Section 7.6.1, for the proper format.

Error Message	Code #
M-Wheel P. False	01 10

Explanation of Diagnostic: This error will occur when requesting Status 46, Length Counter and Status 47, Actual RPM, MW without having a measuring wheel activated in the DLC parameters.

Solution: Refer to Chapter 4 for more information on how to enable External Encoder operation.

Error Message	Code #
Param. # illegal	01 07

Explanation of Diagnostic: The parameter number transmitted to the DLC was too large.

Rexroth Indramat

Solution:

1. Verify that the parameter number transmitted to the DLC is between the following:
   A100 - A125 for Axis parameters
   B000 - B023 for System parameters
   C000 - C012 for Drive parameters
   D01 - D19 for Tool Correction data
   ML00 - ML63 for M-Functions, Low Byte
   MH00 - MH63 for M-Functions, High Byte
2. Refer to Chapter 7, Section 7.6.2, for the proper format.

Error Message	Code #
RS Block # Wrong	01 01

Explanation of Diagnostic: The program command being requested/transmitted to the DLC has an invalid block number.

Solution: Verify that the block number is between 0000 and 2999.

Error Message	Code #
RS Block Data Error	01 03

Explanation of Diagnostic: The format of the program command was transmitted incorrectly to the DLC.

Solution:

1. Verify that the program command format is transmitted correctly to the DLC.
2. Refer to Chapter 7, Serial Interface, for the correct program command format.

Error Message	Code #
RS Checksum Error	01 04

Explanation of Diagnostic: Information transmitted to the DLC with a checksum must agree with the checksum calculated by the DLC. The checksum is used to verify that the information transmitted is correct and complete. The checksum may be enabled/disabled in the Serial Interface parameter, B004.

Solution: Refer to Chapter 7, Section 7.5, for more information on calculating the checksum.

Error Message	Code #
RS Format Error	01 02

Explanation of Diagnostic: The information transmitted contains more characters than is allowed, or a carriage return/line feed is not transmitted at the end of the information.

Solution:

1. Verify that the information transmitted is in the proper format with a carriage return/line feed.
2. Refer to Chapter 7, Serial Interface, for more information on the correct format.

Error Message	Code #
RS-Par.No. False	01 06

Explanation of Diagnostic: An incorrect parameter number was transmitted to the DLC.

Solution:

1. Verify that the parameter number that was transmitted to the DLC was formatted properly.
2. Refer to Chapter 7, Section 7.6.2, for the proper format.

Error Message	Code #
RS-Status# False	01 08

Explanation of Diagnostic: An incorrect status number was transmitted to the DLC.

Solution:

1. Verify that the status number transmitted to the DLC is in the correct format.
2. Refer to Chapter 7 for the correct status numbers and format.

Error Message	Code #
Status # Illegal	01 09

Explanation of Diagnostic: This error occurs if the status number being transmitted to the DLC is not a valid status number.

Solution:

1. Verify that the proper status number is being transmitted to the DLC.
2. Refer to Chapter 7 for the correct status numbers and format.

8.4 G-Code Programming Errors

Error Message	Code #
G01 without F	23

Explanation of Diagnostic: If a G01 command is called up in the DLC program before a feedrate has been programmed using the "F==" command, this error will result. When a G01 command is used, the desired feedrate must be programmed using the "F==" command. Any other feedrate command (ie. VCC, VCA, VEO, etc.) will give you this error.

Solution:

1. Program the desired feedrate using the "F==" command.
2. Verify that no other velocity change commands are programmed between the "F==" and the "G01".

Error Message	Code #
M==CommandStatus # Illegal	20

Explanation of Diagnostic: An M-Function was called for in the program, but M-Function are not properly set up in parameter B009.

When M-Functions are set up in the DLC parameters, parameter B009 requires an input and an output number be specified as the first output used and the first acknowledgment input used. If no input and output are specified, such as the input or output number is 00, when you call for an M-Function, the DLC gives you this error.

Solution: Refer to Chapter 4, Section 4.6 for a complete description of M-Functions and how they can be used.

Error Message	Code #
M==Command M0/1	21

Explanation of Diagnostic: An M-Function acknowledgment was programmed, but M-Function acknowledgments are not properly set up in parameter B009.

When M-Functions are set up in the DLC parameters, parameter B009 requires an input number be specified as the first acknowledgment input. If no input is specified as the starting input, such as the input number is 00, when you call for an acknowledgment in an M-Function, the DLC has not been told what input to monitor and therefore gives you this error.

Solution: Refer to Chapter 4, Section 4.6 for a complete description of M-Functions and how they can be used.

Error Message	Code #
Malfunction S01 Command	22

Explanation of Diagnostic: The Tool Correction register number specified in the current program is not between 01 - 19.

The amount of the correction value to be stored exceeds the value 99999.99 (for 2 decimal place systems) or 9999.999 (for systems set up with 3 decimal places).

Solution:

1. Verify that the value being entered into the Tool Correction register using the SO1 command is valid.
2. See Chapter 5, Section 5.10 for a complete description of Tool Correction register use and programming.
3. See Chapter 5, Section 5.9 for a complete description of the SO1 command and its use. “Refer to Chapter 4, Section 4.6 for a complete description of M-Functions and how they can be used.

Error Message	Code #
No G60	24

Explanation of Diagnostic: There was no G60 command instruction after more than 20 blocks in a G61 Velocity Rate Optimization profile.

A G61 Velocity Rate Optimization profile can cover a maximum of 20 blocks. After 20 blocks, a G60 command must be programmed.

Solution:

1. Insert a G60 block within 20 blocks of the G61 command.
2. Refer to Chapter 5 for a full description of the use of G60 and G61 commands.

CODE #	Diagnostic	Page #
00	(No diagnostic message is displayed)	8-3
45	Abs. Range	8-3
48	Axis 1 Not Homed	8-4
16	Axis No. False	8-4
04	Battery Is Low	8-5
0D	BCD Input	8-5
01 12	Block # Too Large	8-19
1A	Configuration	8-6
07	Division By Zero	8-6
1B	DLC Watch-Dog	8-6
1C	DPR Error	8-7
40	Drive 1 Not Ready	8-7
102	Drive Enabled	8-1
103	Drive Halt	8-1
101	Drive Ready	8-2
42	Drive Runaway A1	8-8
03	EMERGENCY STOP	8-8
4C	Encoder Fault	8-9
43	Excess Pos Lag 1	8-9
41	Feed Angle Loss 1	8-10
23	G01 without F==	8-24
4B	Home Switch Pos	8-10
12	IDS Break Error	8-11
13	IDS Checksum Error	8-11
15	IDS Data Too Long	8-12
14	IDS Format Error	8-12
0E	Invalid Block #	8-13
08	Invalid Mode!	8-13
01 05	Invalid Mode!	8-20
01 11	Invalid P.-Block	8-20
01 13	Invalid Prg Command	8-20
0A	Invalid Prg Command	8-14
02	Is INVALID	8-14
0B	JSR Nesting	8-15
01 10	M-Wheel P. False	8-21
20	M==Command	8-24
21	M==Command M0/1	8-25
22	Malfunction SO1 Command	8-25
47	Max Travel Lmt 1	8-15
46	Min Travel Lmt 1	8-15
24	No G60	8-26

CODE #	Diagnostic	Page #
100	Operation Ready	8-2
01 07	Param. # illegal	8-21
01	Parameter Mode	8-16
05	Parameters Lost	8-16
06	Program lost	8-17
01 01	RS Block # Wrong	8-21
01 03	RS Block Data Error	8-22
01 04	RS Checksum Error	8-22
01 02	RS Format Error	8-22
01 06	RS Par.No. False	8-23
01 08	RS Status # False	8-23
0C	RTS Nesting	8-17
104	Safety Interlock	8-2
18	Software Combination	8-18
01 09	Status # Illegal	8-23
11	System Fault “IRQOVL Interrupt”	8-18
10	System Fault “Stack Overflow” TC Memory Lost!!	8-18
0F	WRITE Command	8-19

Table 8-1: DLC-A Diagnostics Sorted In Alphabetical Order

CODE #	Diagnostic	Page #
00	(No diagnostic message is displayed)	8-3
01	Parameter Mode	8-16
01 01	RS Block # Wrong	8-21
01 02	RS Format Error	8-22
01 03	RS Block Data Error	8-22
01 04	RS Checksum Error	8-22
01 05	Invalid Mode!	8-20
01 06	RS Par.No. False	8-23
01 07	Param. # illegal	8-21
01 08	RS Status # False	8-23
01 09	Status # Illegal	8-23
01 10	M-Wheel P. False	8-21
01 11	Invalid P.-Block	8-20
01 12	Block # Too Large	8-19
01 13	Invalid Prg Command	8-20
02	Is INVALID	8-14
03	EMERGENCY STOP	8-8
04	Battery Is Low	8-5
05	Parameters Lost	8-16
06	Program lost	8-17
07	Division By Zero	8-6
08	Invalid Mode!	8-13
0A	Invalid Prg Command	8-14
0B	JSR Nesting	8-15
0C	RTS Nesting	8-17
0D	BCD Input	8-5
0E	Invalid Block #	8-13
0F	WRITE Command	8-19
10	System Fault “Stack Overflow”	8-18
100	Operation Ready	8-2

CODE #	Diagnostic	Page #
101	Drive Ready	8-2
102	Drive Enabled	8-1
103	Drive Halt	8-1
104	Safety Interlock	8-2
11	System Fault “IRQOVL Interrupt”	8-18
12	IDS Break Error	8-11
13	IDS Checksum Error	8-11
14	IDS Format Error	8-12
15	IDS Data Too Long	8-12
16	Axis No. False	8-4
18	Software Combination	8-18
1A	Configuration	8-6
1B	DLC Watch-Dog	8-6
1C	DPR Error	8-7
20	M==Command	8-24
21	M==Command M0/1	8-25
22	Malfunction SO1 Command	8-25
23	G01 without F==	8-24
24	No G60	8-26
40	Drive 1 Not Ready	8-7
41	Feed Angle Loss 1	8-10
42	Drive Runaway A1	8-8
43	Excess Pos Lag 1	8-9
45	Abs. Range	8-3
46	Min Travel Lmt 1	8-15
47	Max Travel Lmt 1	8-15
48	Axis 1 Not Homed	8-4
4B	Home Switch Pos	8-10
4C	Encoder Fault	8-9

Table 8-2: DLC-A Diagnostics Sorted by Code #

A Programming Notes

This section is periodically updated with hints and examples of use and application of different programming commands.

Notes included in this section:

Page	Description
A-1	Axis Homing for the DLC

A.1 Axis Homing for the DLC

General

Homing of a linear or rotary axis is required if you need to do absolute positioning and are using an incremental position feedback device mounted on the motor or machine. The homing process, usually a series of back and forth moves to locate a reference point in relation to a mechanical setup, can be accomplished through a variety of methods when using the DLC control. This Programming Note describes several different homing methods in detail.

Note: In some applications, for mechanical or safety reasons, the axis cannot be homed after the process has started. For these applications, a multi-turn absolute encoder is required. Because an absolute device is capable of providing a known position relative to the machine's mechanical limits at all times, homing is not required.

Normal Homing

The DLC includes a homing routine suitable for most applications. This is described in detail in section 3.2.4 of the DLC manual. This routine, in conjunction with parameters and a home switch, provides a fast and easy way for the user to implement a homing sequence. This routine is activated by an input in manual mode or through a short series of commands in automatic mode. This home sequence in either mode starts with the selected axis performing the following steps.

1. The axis moves towards the home switch, at the velocity set in parameter A110, unless the axis is already on the switch.
2. After the home switch closes, the axis decelerates to a stop on the marker pulse.

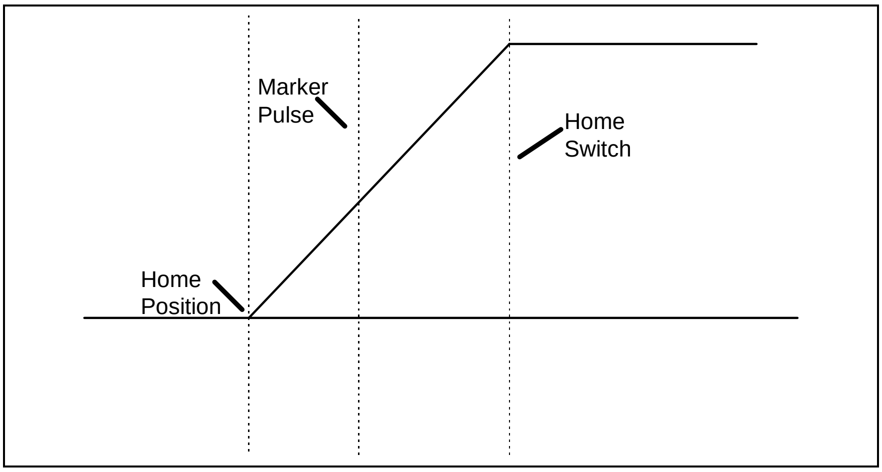


Fig. A-1: Normal Homing Example

Rexroth Indramat

Homing Without Using the Homing Routine

In certain applications, it is not possible or desirable to use the standard homing sequence. For those instances, the user must define a homing sequence that satisfies the application, then write a short program routine to home the axes. With this method, the user can customize the homing routine to compensate for backlash, forward-moving-only applications, homing to a switch, or a variety of other needs.

Homing to a Switch

The following program sequences demonstrate several methods of homing to a switch. The first example is used to home an axis that cannot backup. The second example will provide a more accurate homing relative to the home switch, because the axis backs up to the switch. The third example is used if you must detect whether the axis is already on the home switch and must back off the switch before homing. Note that when customized homing routines are used, they are only executed in automatic mode, and the ability to home in manual mode is lost.

Note: The following examples assume the parameters and I/O are set as shown below. This can be freely modified to fit the particular application.

Inputs		**Outputs**	
1	Home switch, axis 1 (A112)	21	axis 1 Homed (A112)
		23	axis 1 In Position (A106)

Example 1: Illustrates the simplicity of creating a routine to home both axes in sequence (axis 1 first, then axis 2). This routine is very useful for an application in which the axis should not back up while homing, such as a rotary table, conveyor or continuous web.

The program starts with blocks 0000, used to detect if either axis has been previously homed. If not, the homing routines at blocks 0800 will be executed as required. The homing program starts by branching to the CON command, which causes the program to wait until the home switch input is activated. The PBK command causes the axis to ramp to a stop at the current acceleration rate. The distance traveled past the switch is a function of both the selected velocity and the acceleration rate. The ATS command waits for the axis to be fully stopped (in position) before the CLA command is executed. The CLA command sets the axis position buffer to zero and sets the Homed flag high. This completes the homing process.

```
0000     BCA     0800  21   0    ;Branch if axis 1 has not been homed
0001     NOP                     -- Start of Program --

0800     CON     1  1  -050   00 ;Sets axis 1 in (-) direction at 5% max. velocity
0801     AKN     1  1            ;Waits for axis 1 home switch to close
0802     PBK     1               ;Stops axis 1 motion (CON off could also be used)
0803     ATS     23  1           ;Waits for axis 1 to be in position
0804     CLA     1               ;Initializes axis 1 position buffer to zero
0805     JMP     0000
```

Example 2: Illustrates the use of the REF command, which causes the axis to back up to the point where the home switch was first closed before setting the position buffer to zero. Otherwise, it will function similar to example 1.

```
0800     REF     1  1  050  01   ;Sets axis 1 in REV direction, at 5% max. velocity
                                 ;waits for home switch to close, then ramps to stop
                                 ;and reverses to the point the switch first closed.
0801     ATS     23  1           ;Waits for axis 1 to be in position
0802     CLA     1               ;Initializes axis 1 position buffer to zero
0803     NOP
0804     JMP     0000
```

Example 3: Illustrates a routine that is very useful for applications in which the axis should not back up past the home switch. The program starts with blocks 0000, used to detect if either axis has been previously homed. If not, the homing routines at blocks 0800 will be executed as required. The homing program starts by branching to a BCE command that will branch to block 805 (core homing program) if the switch is not closed. If the home switch was closed, the program will drop down to the next three steps, comprised of a CON command which will cause the axis to move off the switch at a set velocity, an AKN command which causes the program to wait until the home switch input is opened, and the PBK which causes the axis to ramp to a stop.

The REF command starts the core homing program, used to locate the home switch. The axis moves toward the home switch at a set velocity until the switch is closed. After this, the axis ramps to a stop and then moves in the reverse direction, stopping at the point where the switch was first closed. The ATS command waits for the axis to be fully stopped (in position) before the CLA command is executed. The CLA command set the axis position buffer to zero and sets the Homed Flag high. This completes the homing process for a given axis.

```
0000   BCA   0800  21   0 ;Branch if axis 1 has not been homed

0800   BCE   0805  1    0       ;Branch if axis 1 is not on the home switch
0801   CON   1  1   +050  00    Sets axis 1 in (+) direction at 5% max. velocity
0802   AKN   1  0               Waits for axis 1 home switch to open
0803   PBK   1                  Stops axis 1 motion
0804   NOP
0805   REF   1  1    050  01    ;Sets axis 1 in REV direction at 5% max. velocity, waits
                                for home switch to close, then ramps to a stop
                                and reverses to the point the switch first closed.
0806   ATS   23  1              Waits for axis 1 to be in position
0807   CLA   1                  Initializes axis 1 position buffer to zero
0808   JMP   0000
```

Homing to the Marker Pulse

When the DLC Servo System is first powered up, it knows where the motor pulse is located. The DLC can tell the axis to home to the marker pulse. This is very useful for single revolution applications where the rotating mechanisms are directly attached to the motor shaft (rotary knife, print drum). **This type of operation is enabled when the home switch input is set to 00 in parameter A112.** The homing routine can be activated by an input in manual mode, or through a short series of commands in automatic mode. This home sequence, in either mode, starts with the axis performing the following steps:

1) The axis moves in the desired direction (CW, CCW) at the velocity set in parameter A110, until the marker pulse is detected.

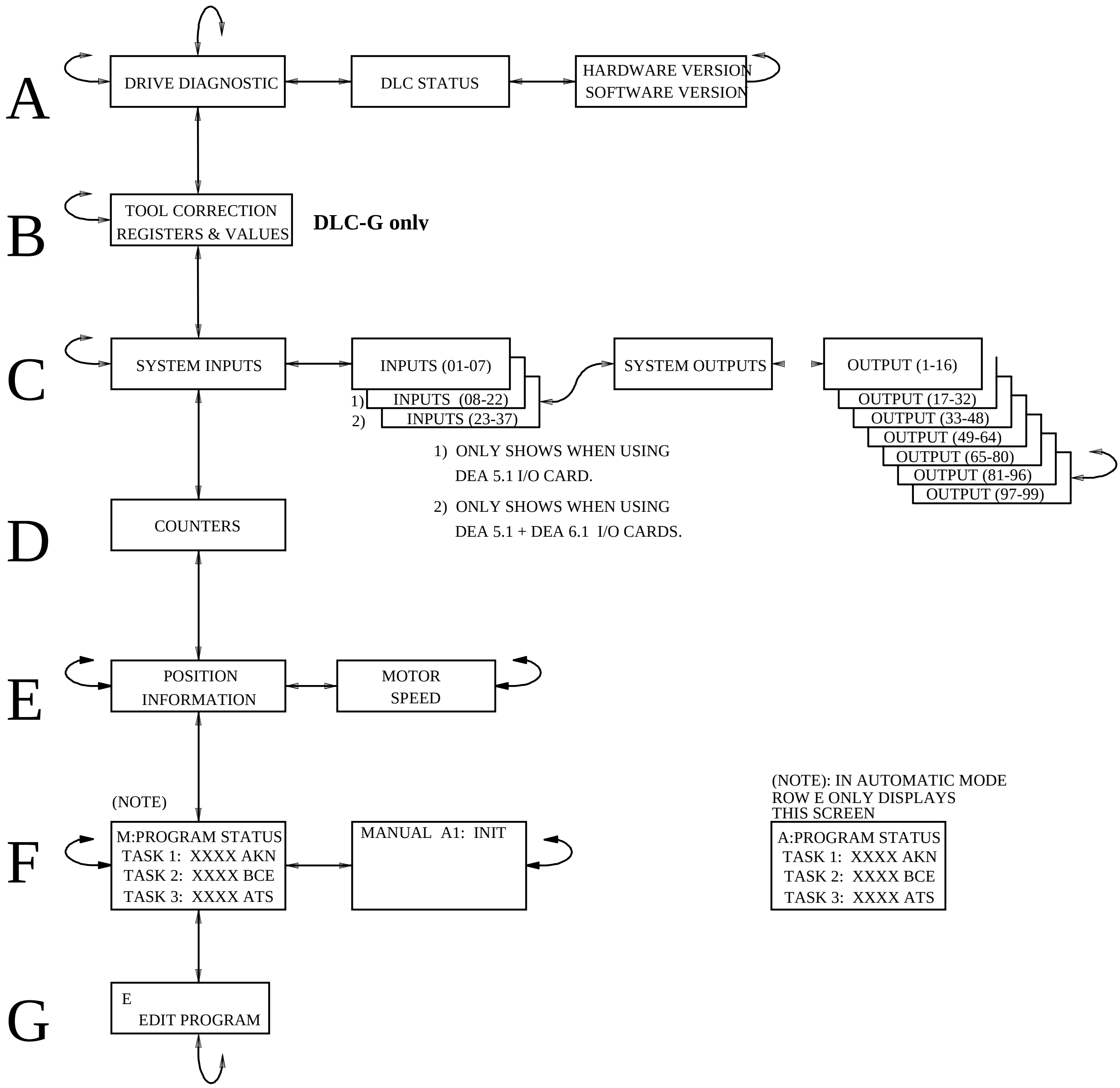
Homing Routine at Start of User Program

```
0000     BCE     005  21  1     ;Branch if axis 1 is already homed
0001     HOM     1              Home axis 1 (figure 2)
0002     ATS     21    1 Wait for axis 1 Homed output
0003     JMP     0010

0010     NOP                    ;      -- Start of Program --
0011
```

B DLC Display Screen Map

This page is included for illustrative purposes only. Refer to chapter 2, section 2.3, for description of using this map, and how to interpret the information on the screens of the DLC control panel display.



C Drawings and Schematics

CAUTION: The drawings in this Appendix are included for illustrative purposes only and are subject to change without notice. Check with Indramat to be sure you are working with the latest drawings prior to installing, wiring and powering equipment.

C.1 DKS/DLC-A/MDD Interconnection (Sheets 1 – 10)

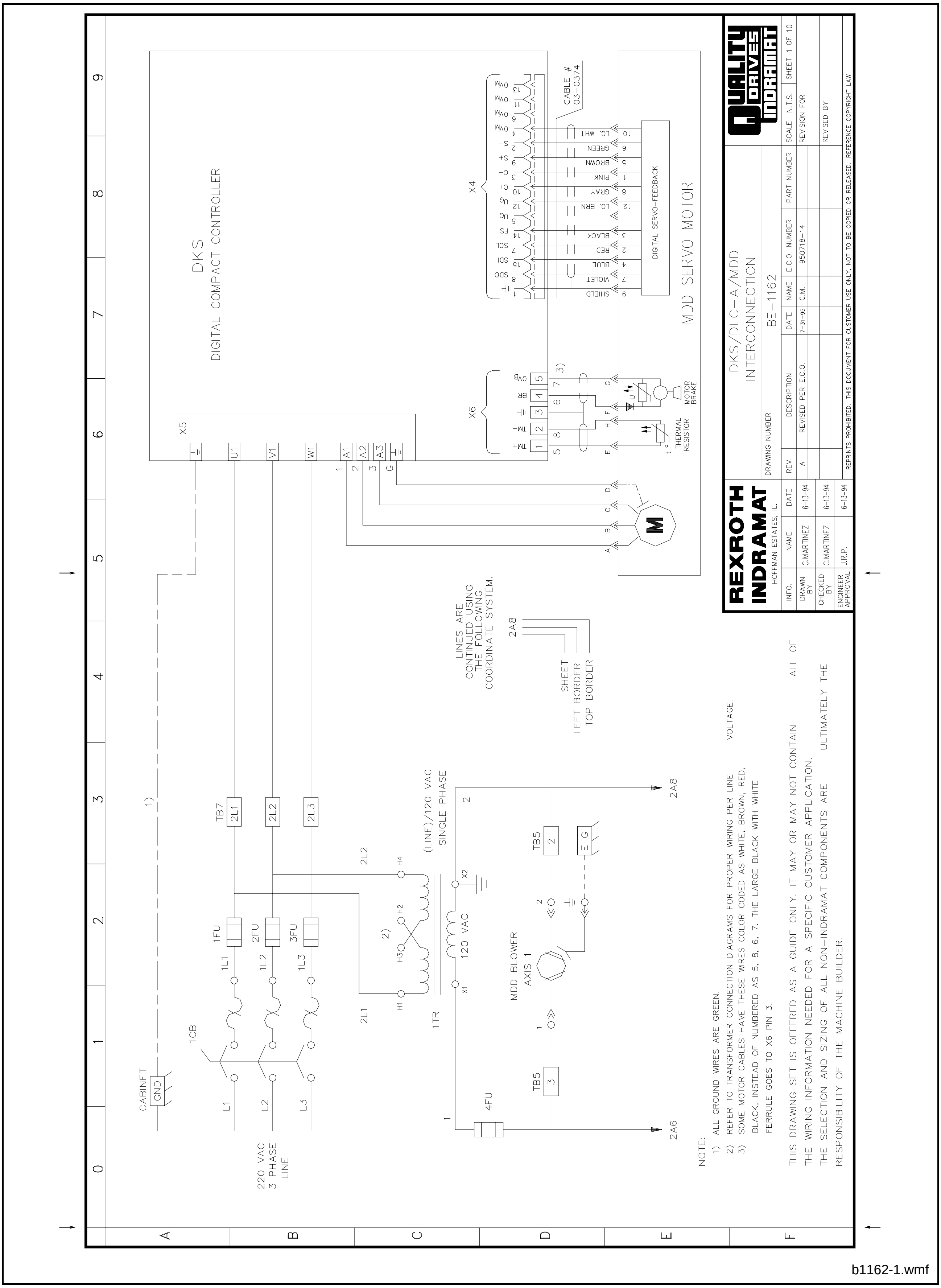
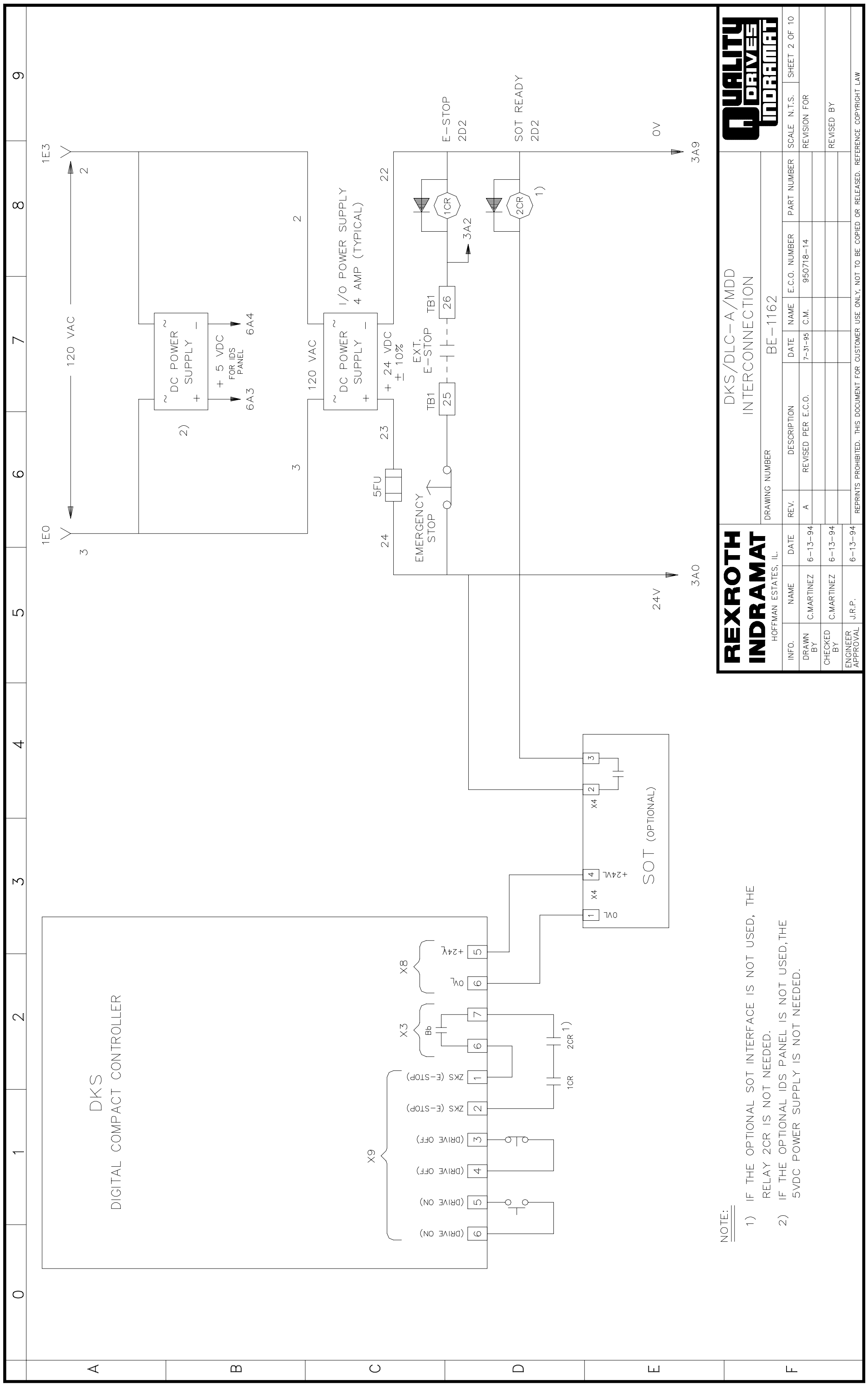
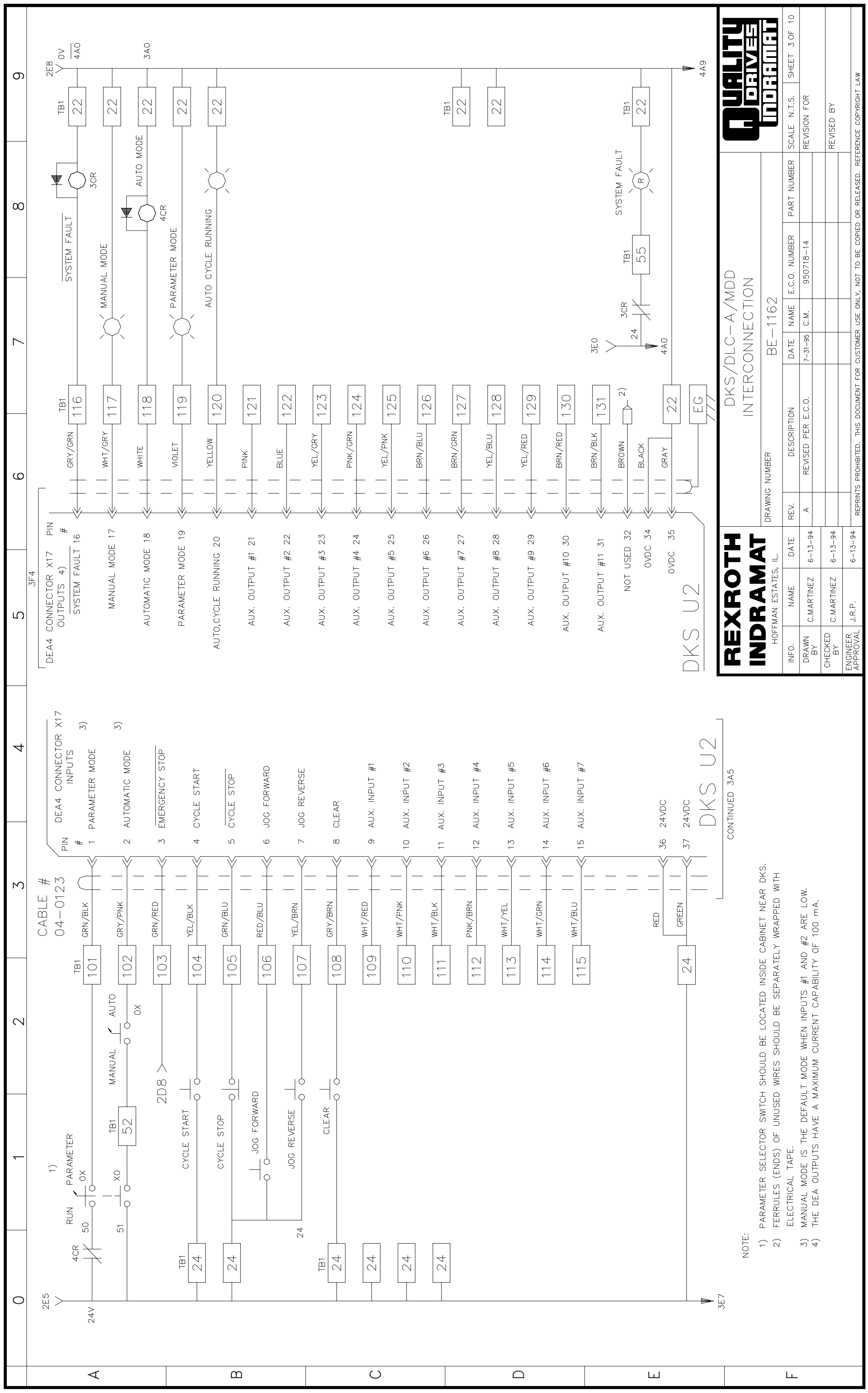


Figure C-1: DKS/DLC-A/MDD Interconnection, Sheet 1 of 10



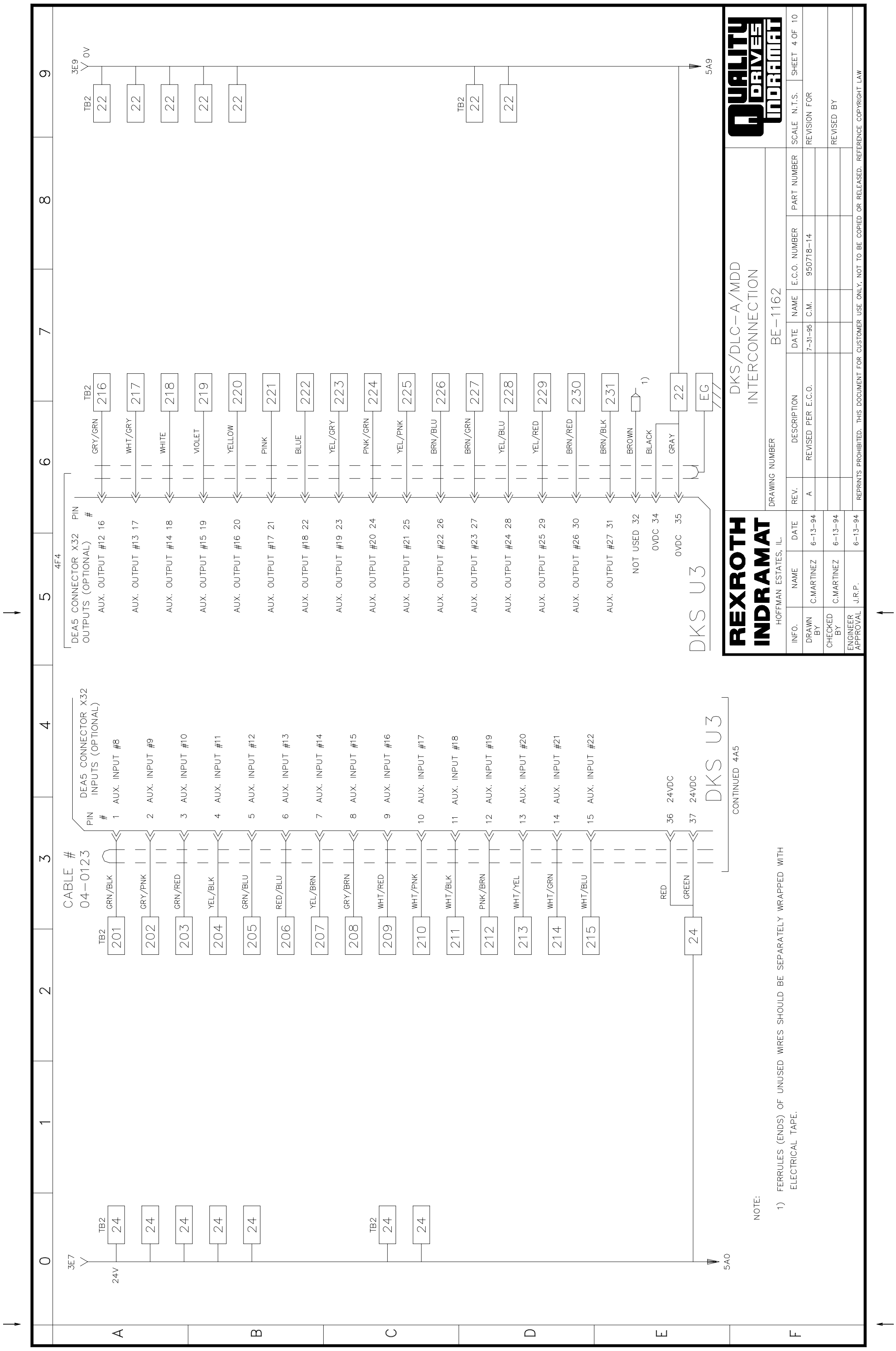
b1162-2.wmf

Figure C-2: DKS/DLC-A/MDD Interconnection, Sheet 2 of 10



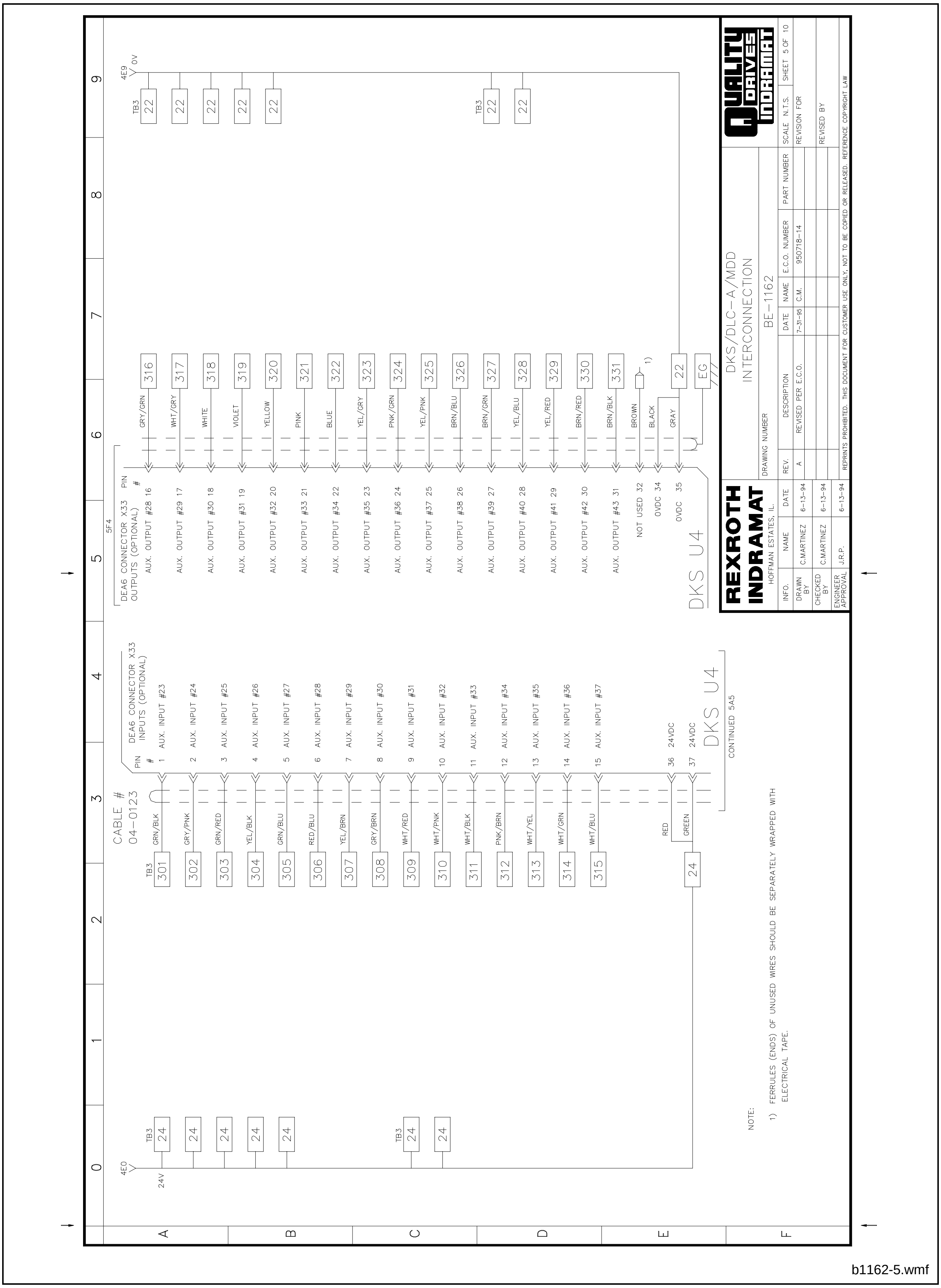
b1162-3.wmf

Figure C-3: DKS/DLC-A/MDD Interconnection, Sheet 3 of 10



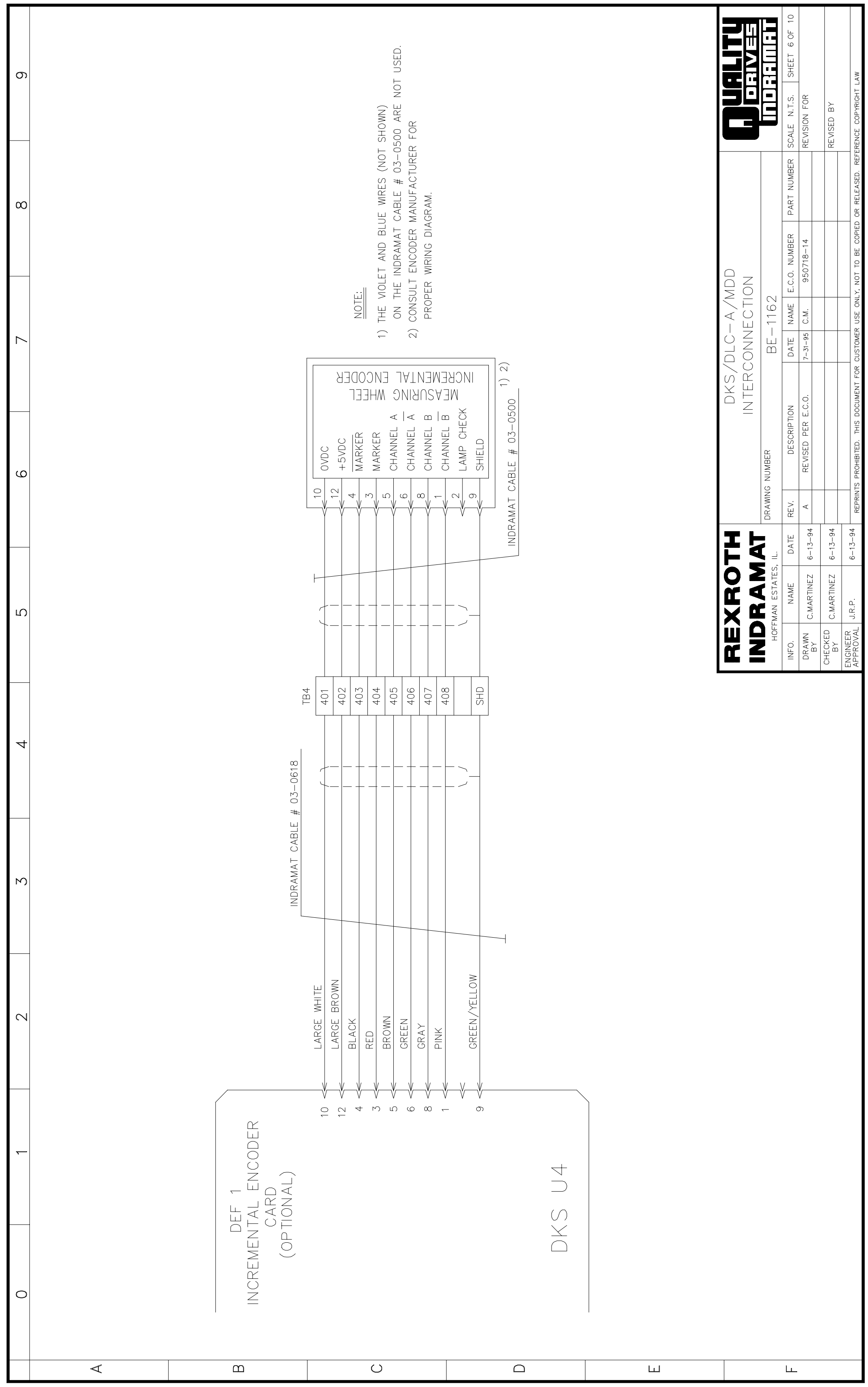
b1162-4.wmf

Figure C-4: DKS/DLC-A/MDD Interconnection, Sheet 4 of 10



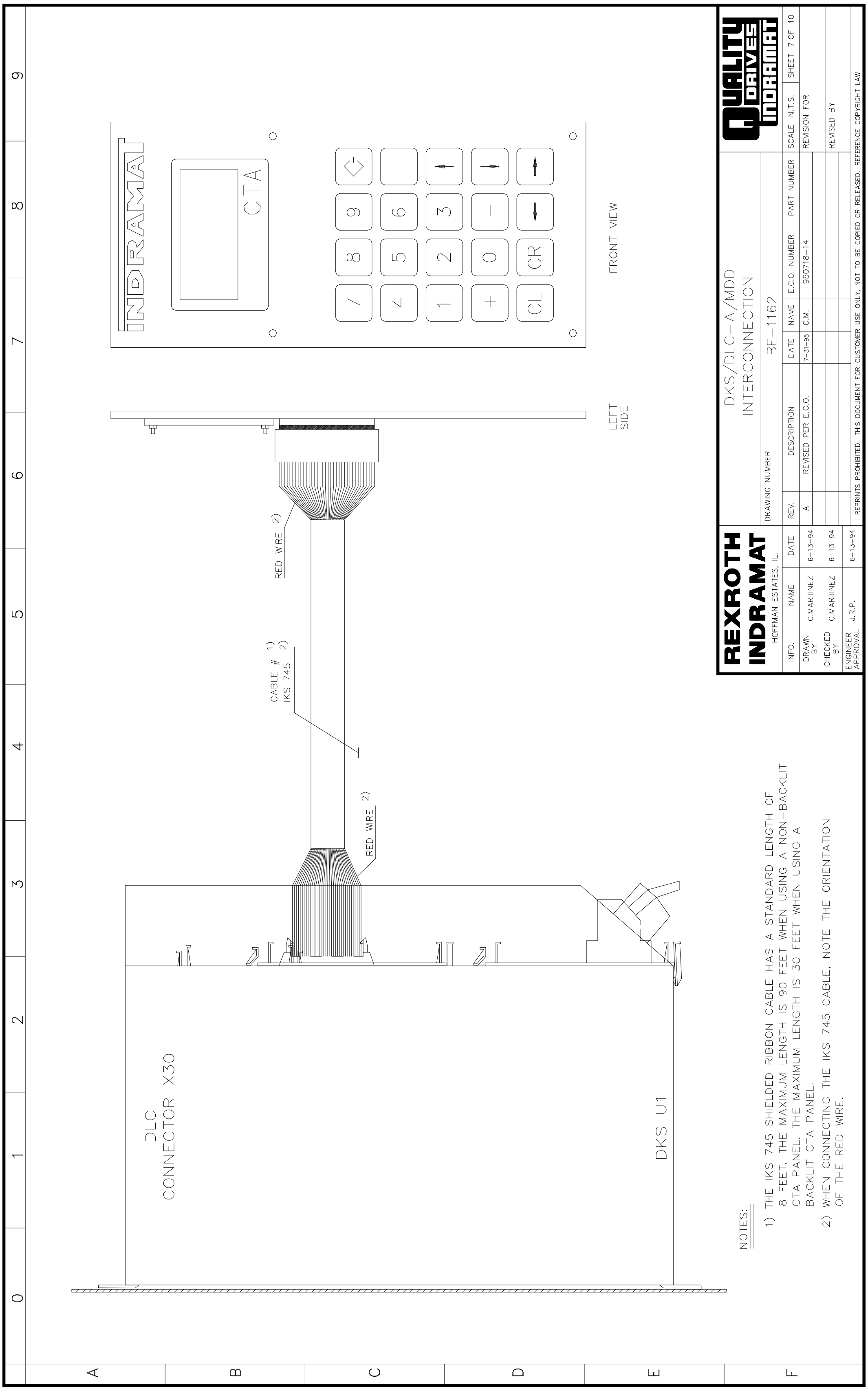
b1162-5.wmf

Figure C-5: DKS/DLC-A/MDD Interconnection, Sheet 5 of 10



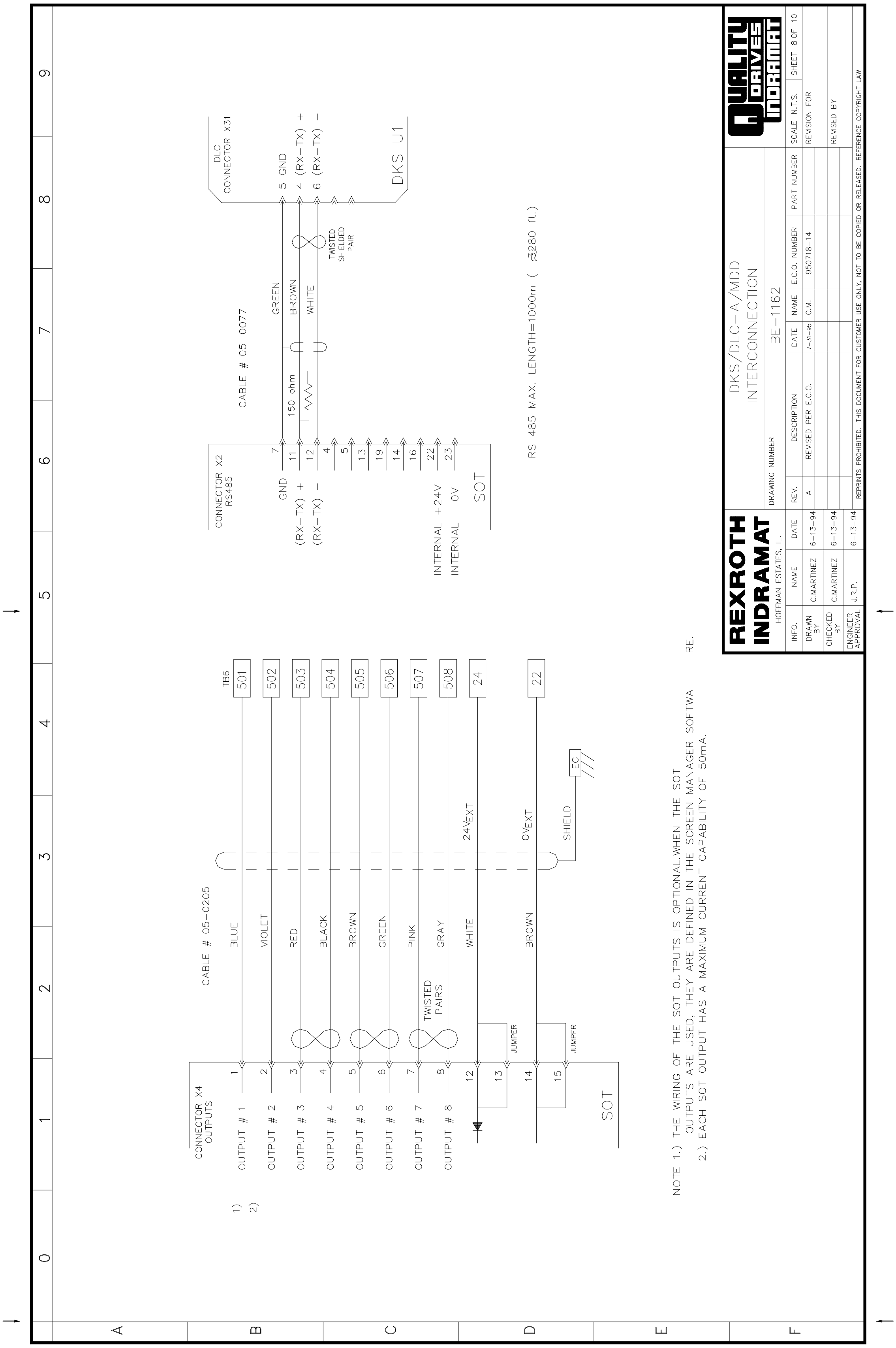
b1162-6.wmf

Figure C-6: DKS/DLC-A/MDD Interconnection, Sheet 6 of 10



b1162-7.wmf

Figure C-7: DKS/DLC-A/MDD Interconnection, Sheet 7 of 10



b1162-8.wmf

Figure C-8: DKS/DLC-A/MDD Interconnection, Sheet 8 of 10

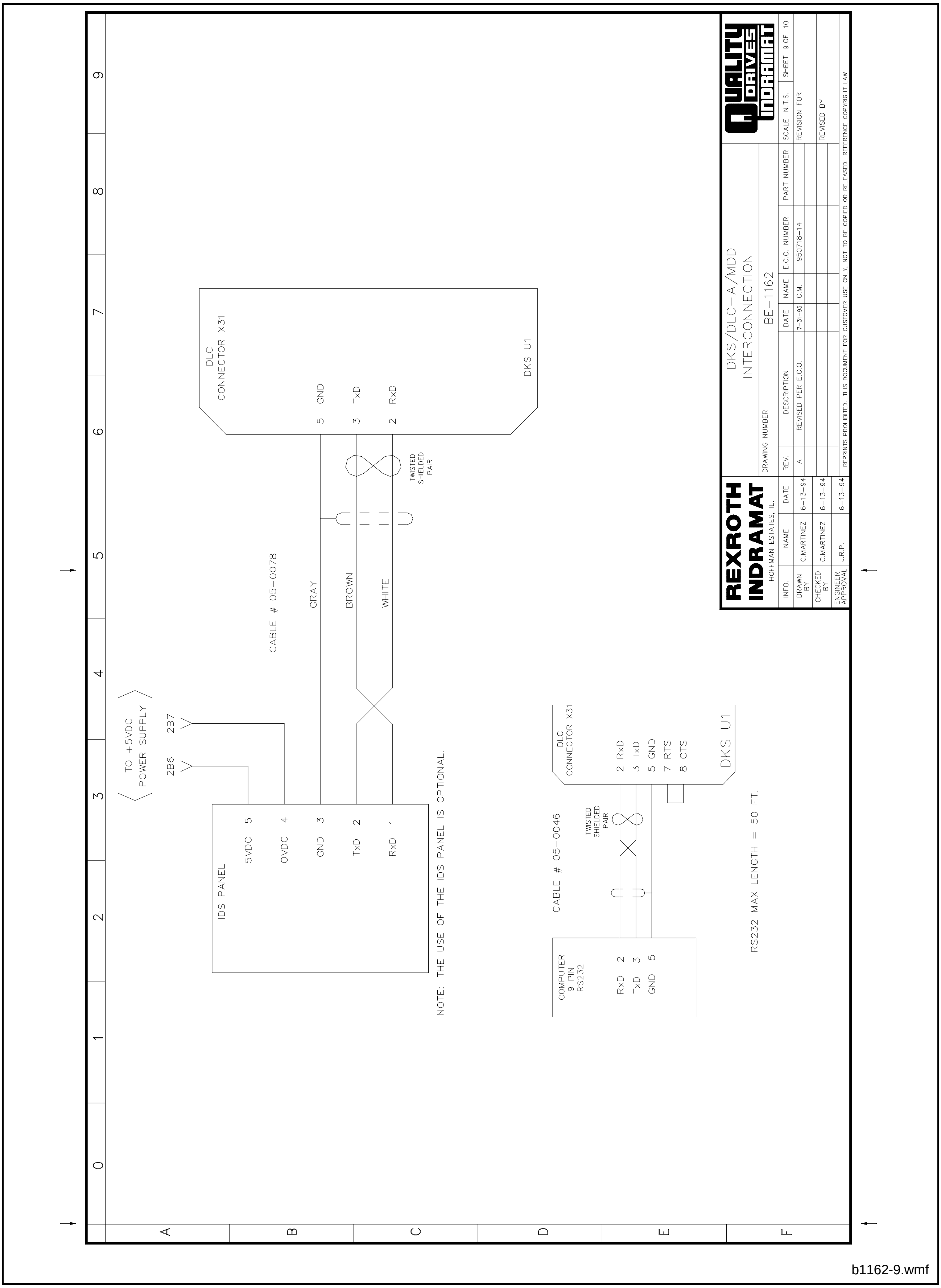
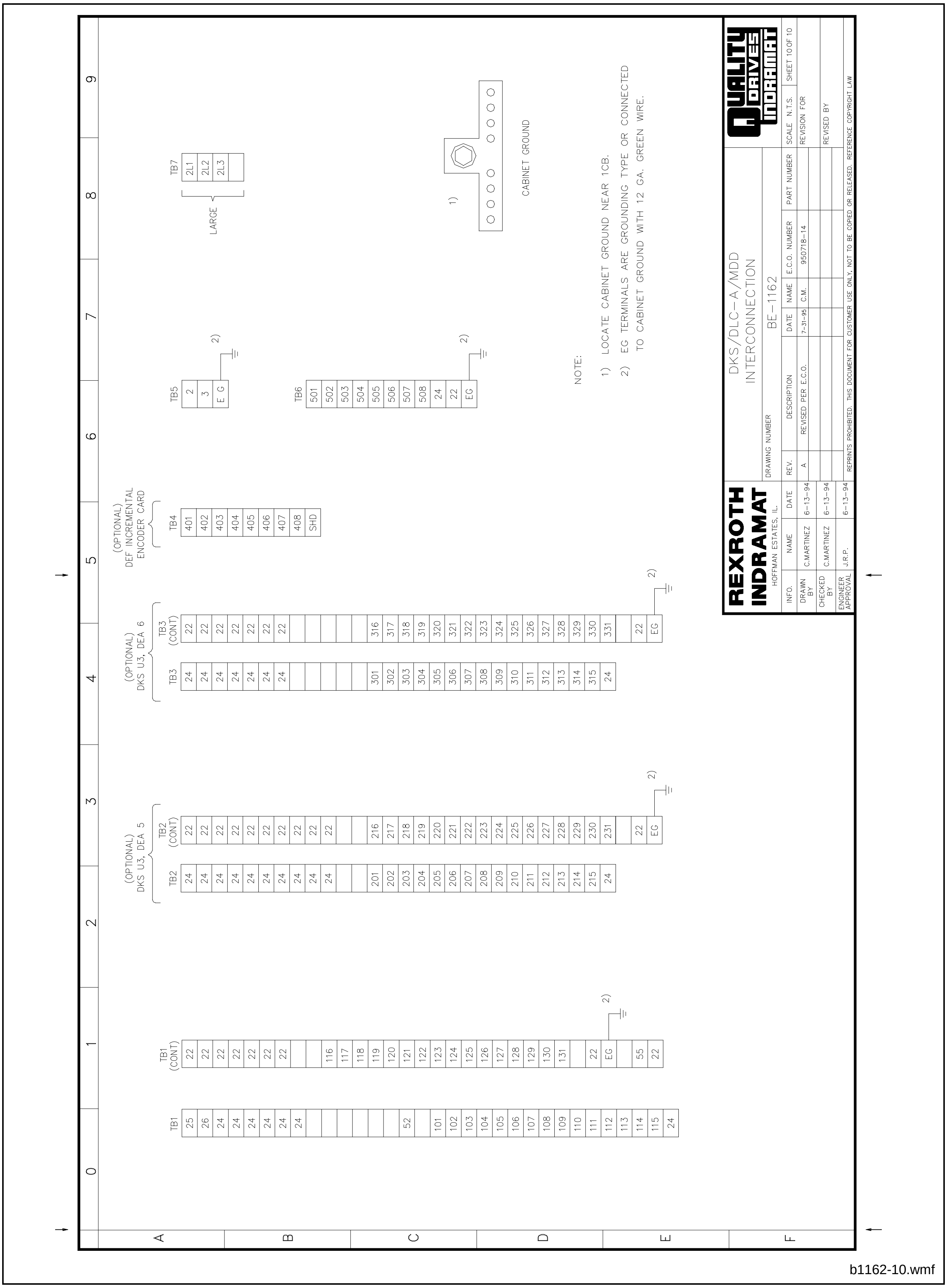


Figure C-9: DKS/DLC-A/MDD Interconnection, Sheet 9 of 10



b1162-10.wmf

Figure C-10: DKS/DLC-A/MDD Interconnection, Sheet 10 of 10

Rexroth Indramat

C.2 RS 232 Data Interface Interconnection

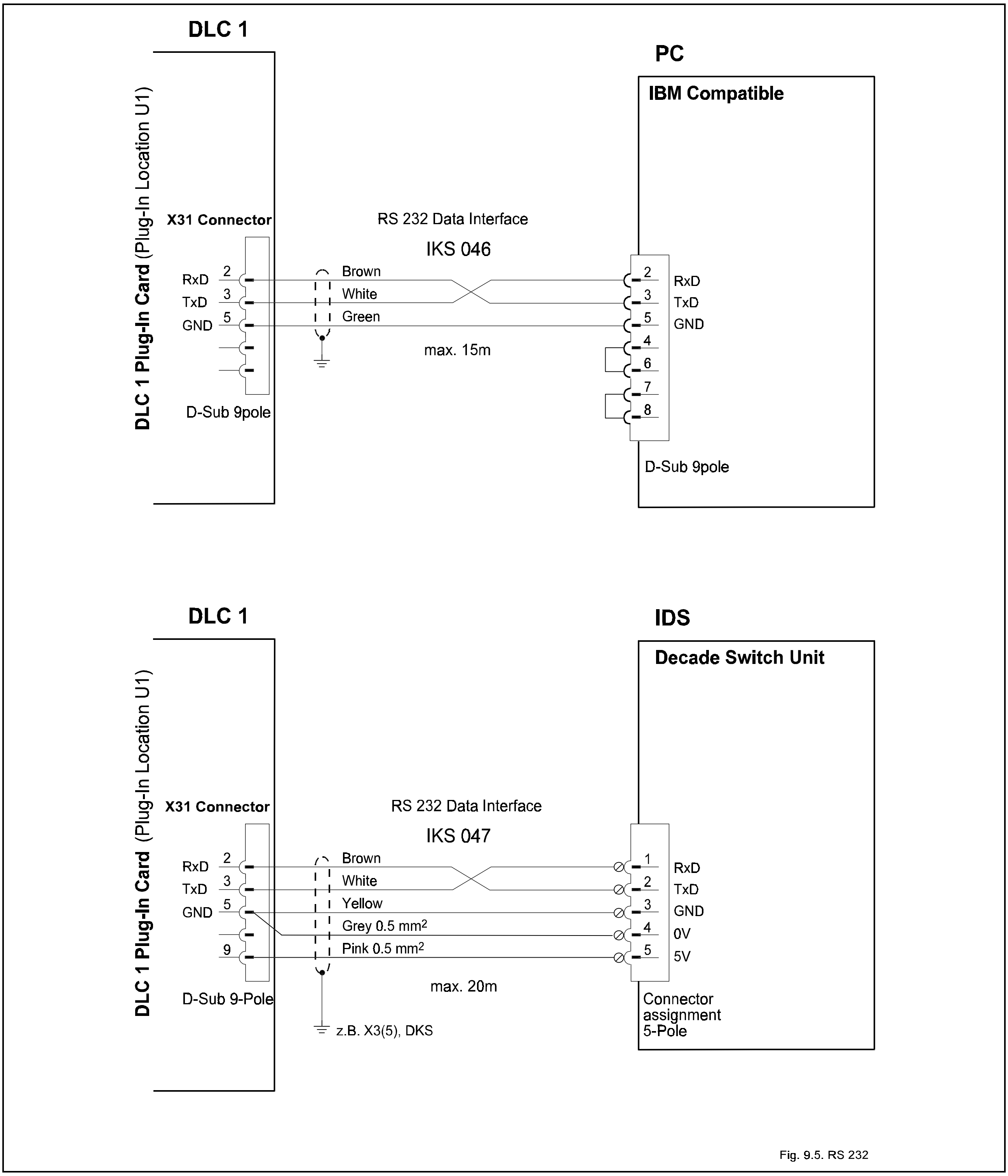
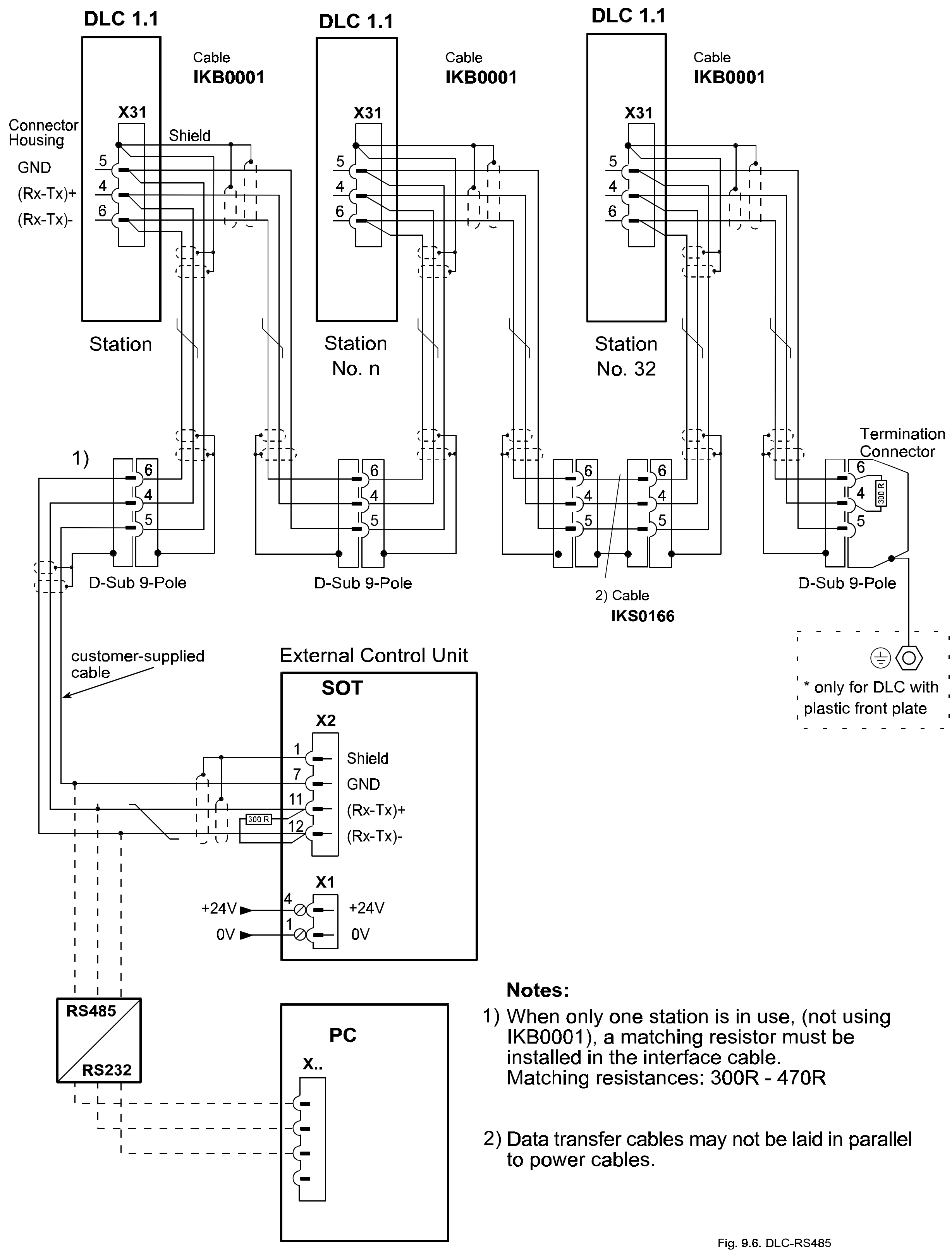


Figure C-11: RS 232 Data Interface Interconnection

C.3 SOT – DLC RS485 Interconnection



Notes:

1) When only one station is in use, (not using IKB0001), a matching resistor must be installed in the interface cable. Matching resistances: 300R - 470R

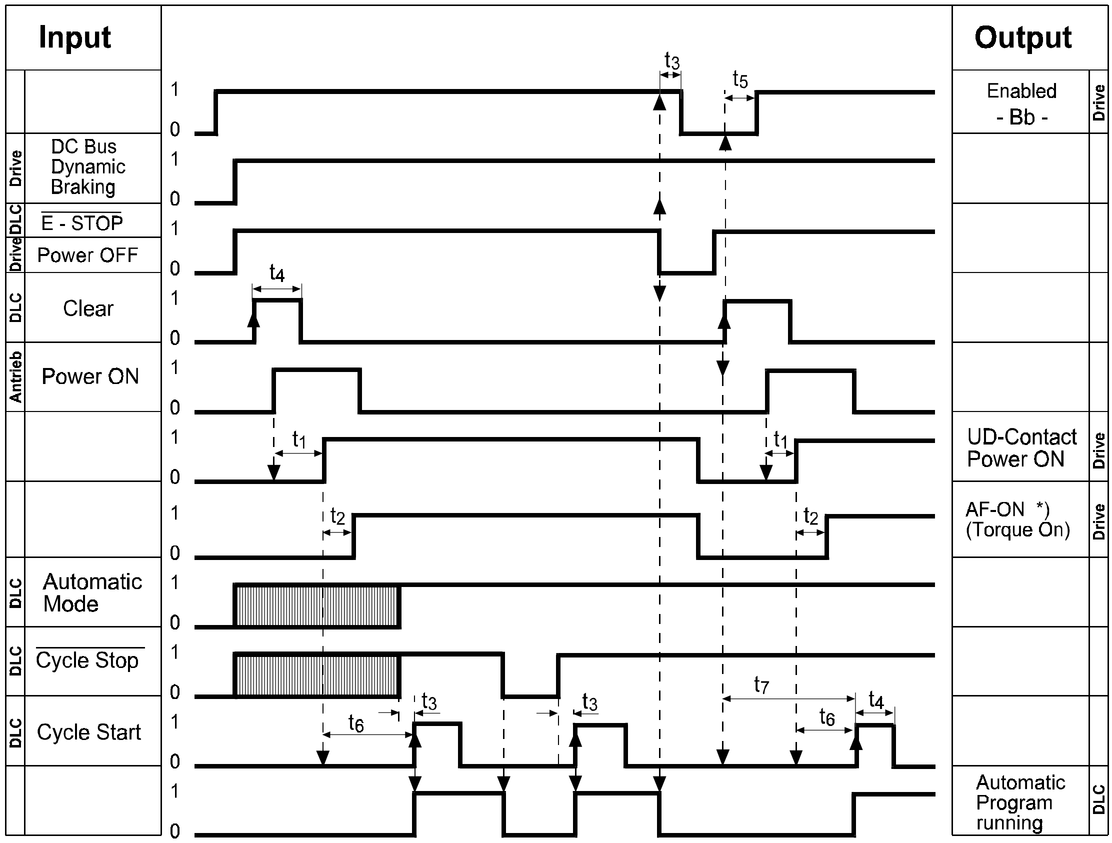
2) Data transfer cables may not be laid in parallel to power cables.

Fig. 9.6. DLC-RS485

fig906.wmf

Figure C-12: SOT – DLC RS485 Interconnection with up to 32 Stations

C.4 Timing Diagram for DLC with DKS, DDS or DDC



t_1 approx. 5s for DKS
approx. 300ms for DDC and DDS with TVD

t_2 >200ms after locking the UD contact and if Bb = `1´,
the drive torque falls.

t_3 ≥5ms

t_4 ≥20ms

t_5 ≥10ms

t_6 ≥300ms after locking the UD contact and if Bb = `1´,
the program can be started.

t_7 Maintain >800ms minimum interval between CLEAR and START after an

*) AF-ON = internal DLC drive enable for the drive controller.
When AF-ON is activated, the drive torque falls.

These timing diagrams enable powering up the DLC and drive controller without errors. To shut down drives in case of an Emergency Stop, see documentation for the drive controller or power supply module (for modular drive controllers).

Fig. 9.7. Tim.

Figure C-13: Timing Diagram for DLC with DKS, DDS or DDC

D Installation Drawings

CAUTION: The drawings in this Appendix are included for illustrative purposes only and are subject to change without notice. Check with Indramat to be sure you are working with the latest drawings prior to installing, wiring and powering equipment.

D.1 DKS 1.1 Dimensions

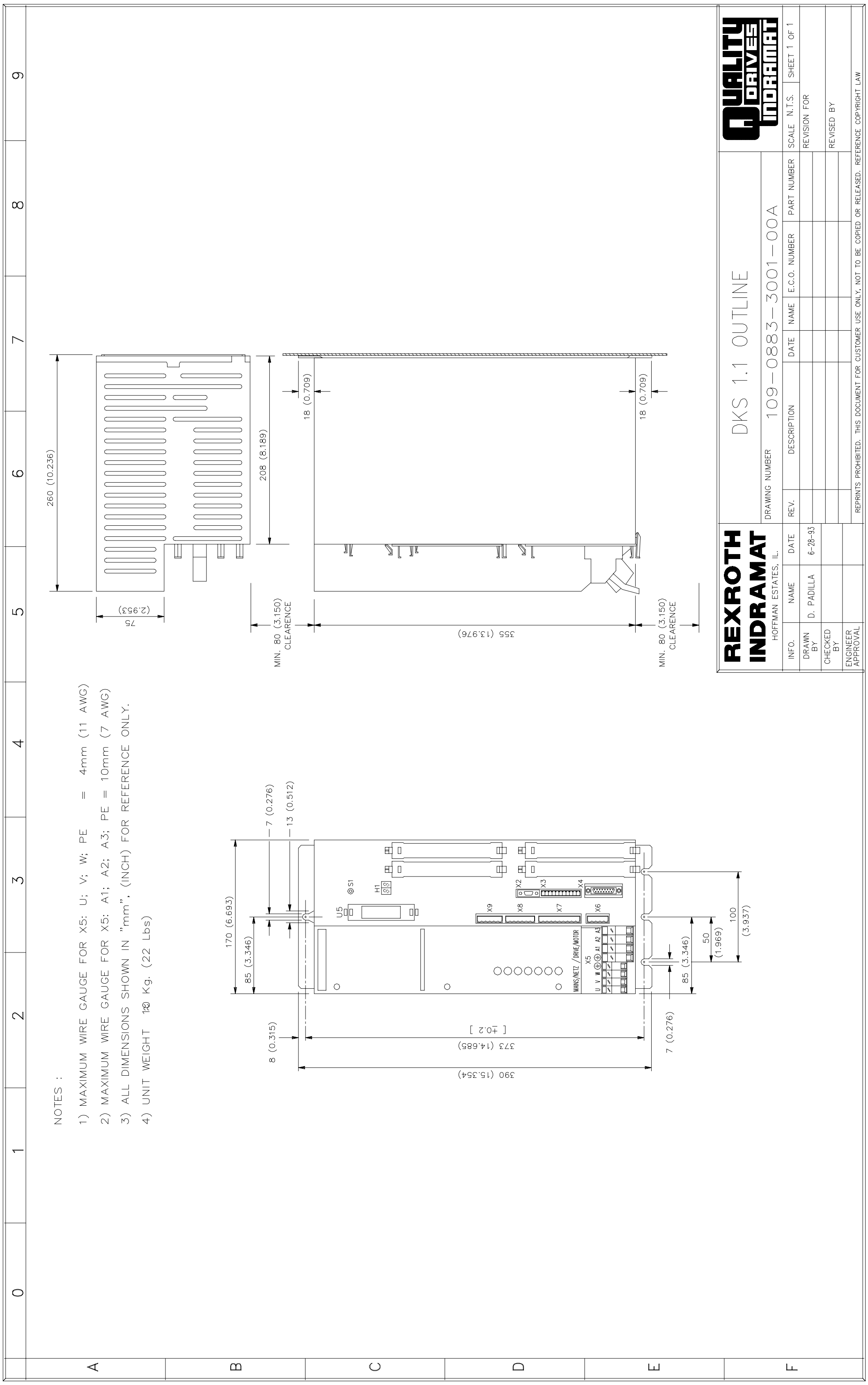
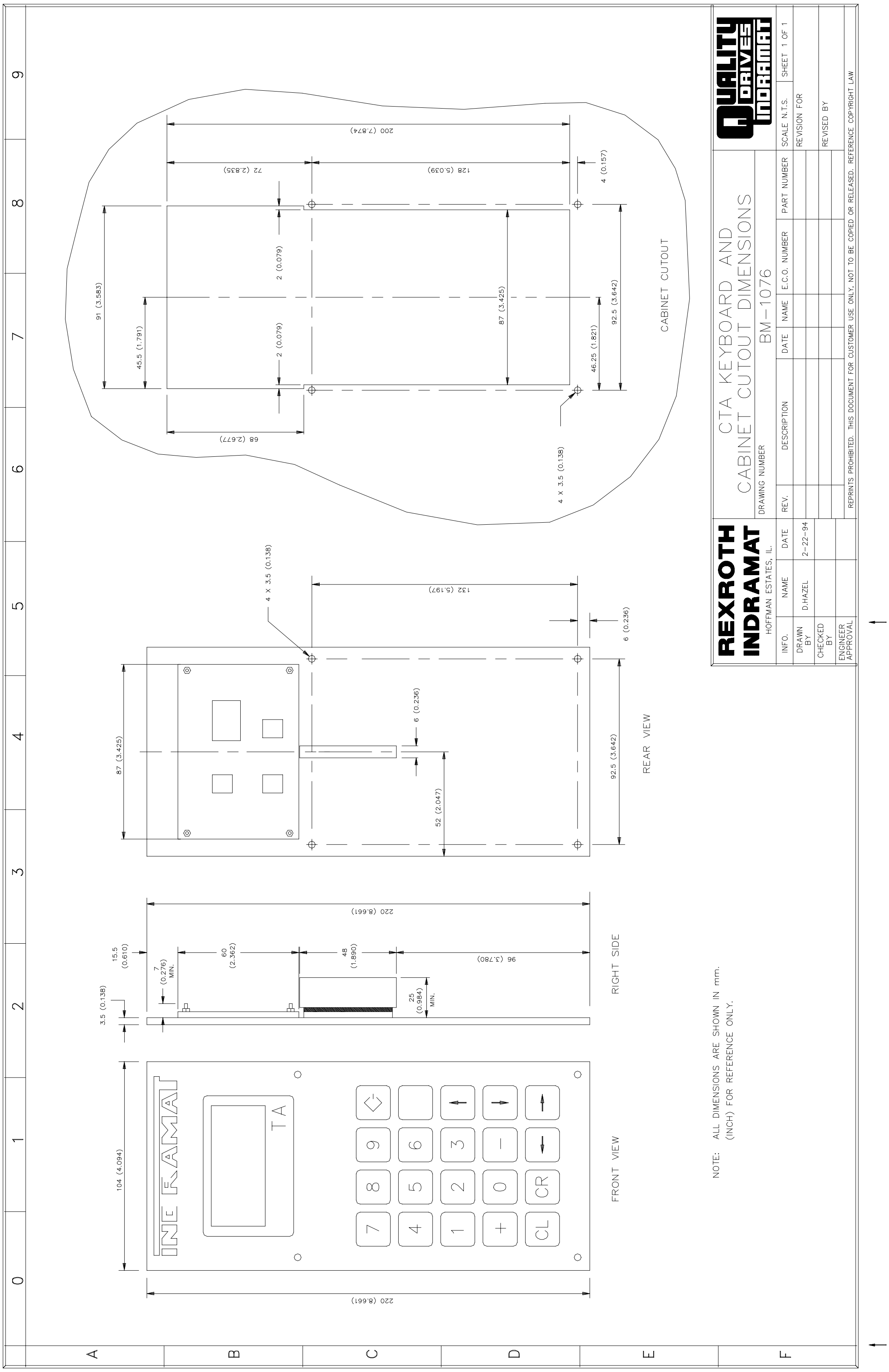

dks1-out.wmf

Figure D-1: DKS 1.1 Dimensions

D.2 CTA Keyboard and Cabinet Cutout Dimensions



bm-1076.wmf

Figure D-2: CTA Keyboard and Cabinet Cutout Dimensions

D.3 CTA 04 Keyboard and Cabinet Cutout Dimensions

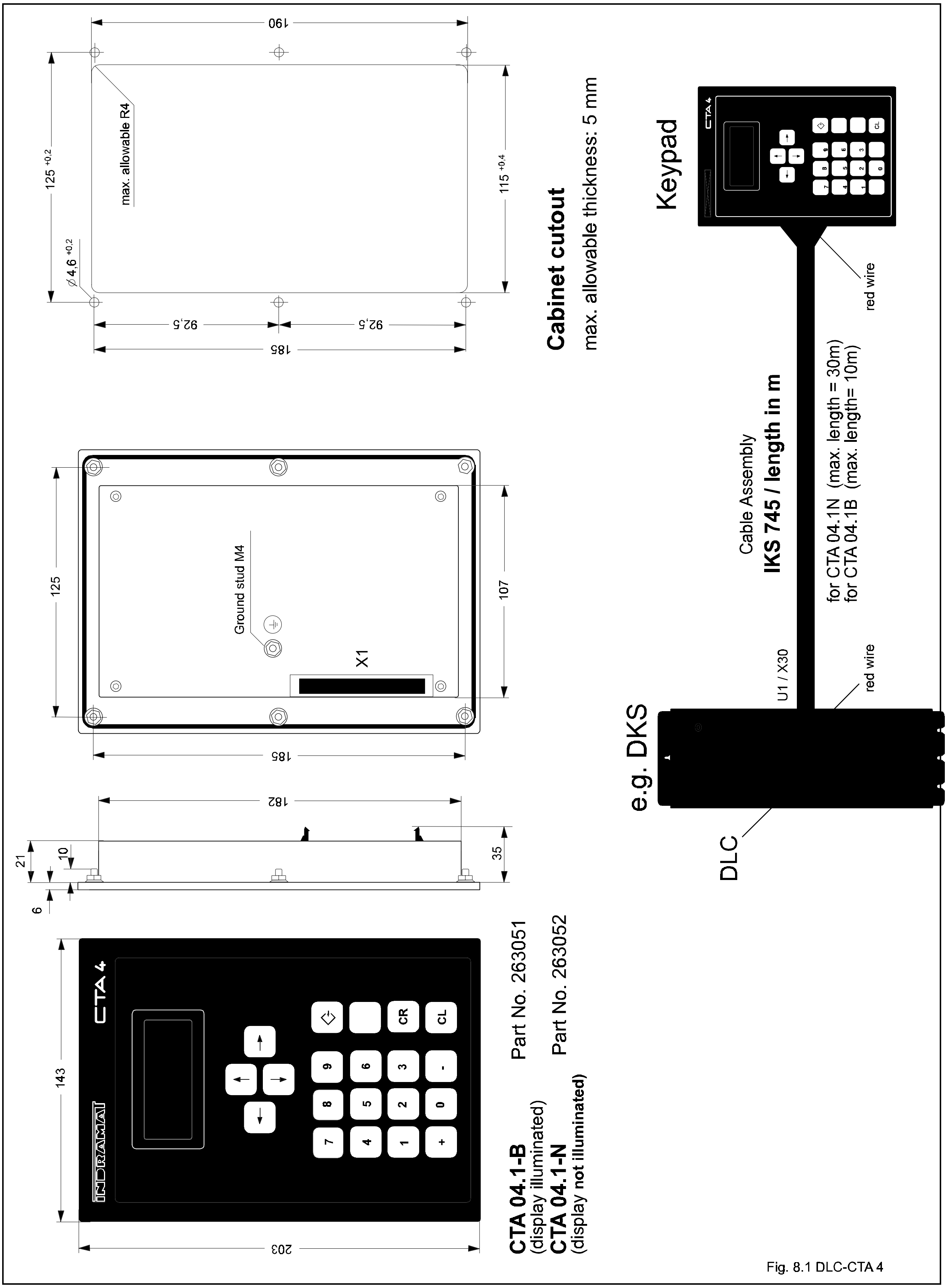


Figure D-3: CTA 04 Keyboard and Cabinet Cutout Dimensions

D.4 IDS and Cabinet Cutout Dimensions

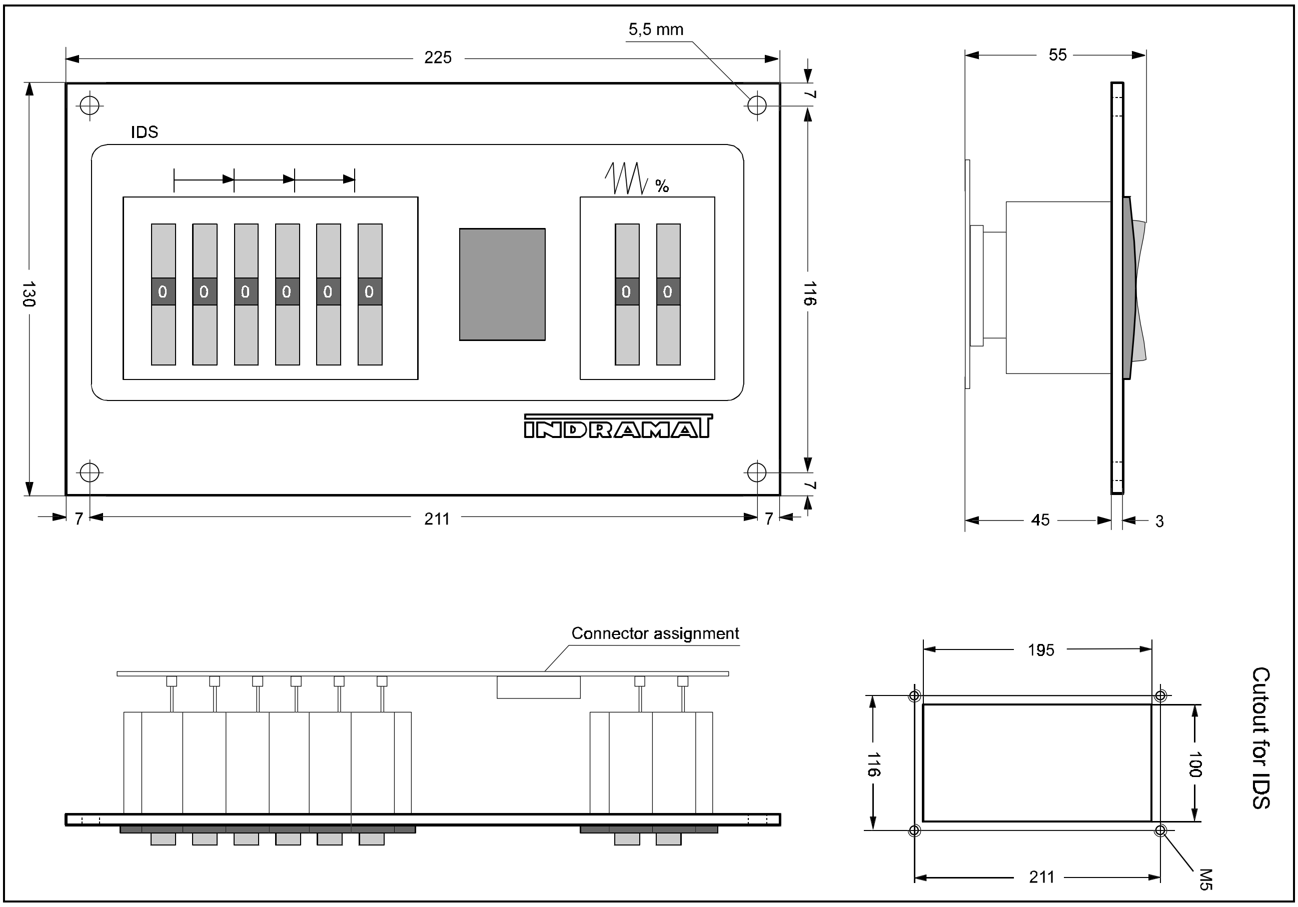


Figure D-4: IDS and Cabinet Cutout Dimensions

D.5 CTA 10 Keyboard and Cabinet Cutout Dimensions

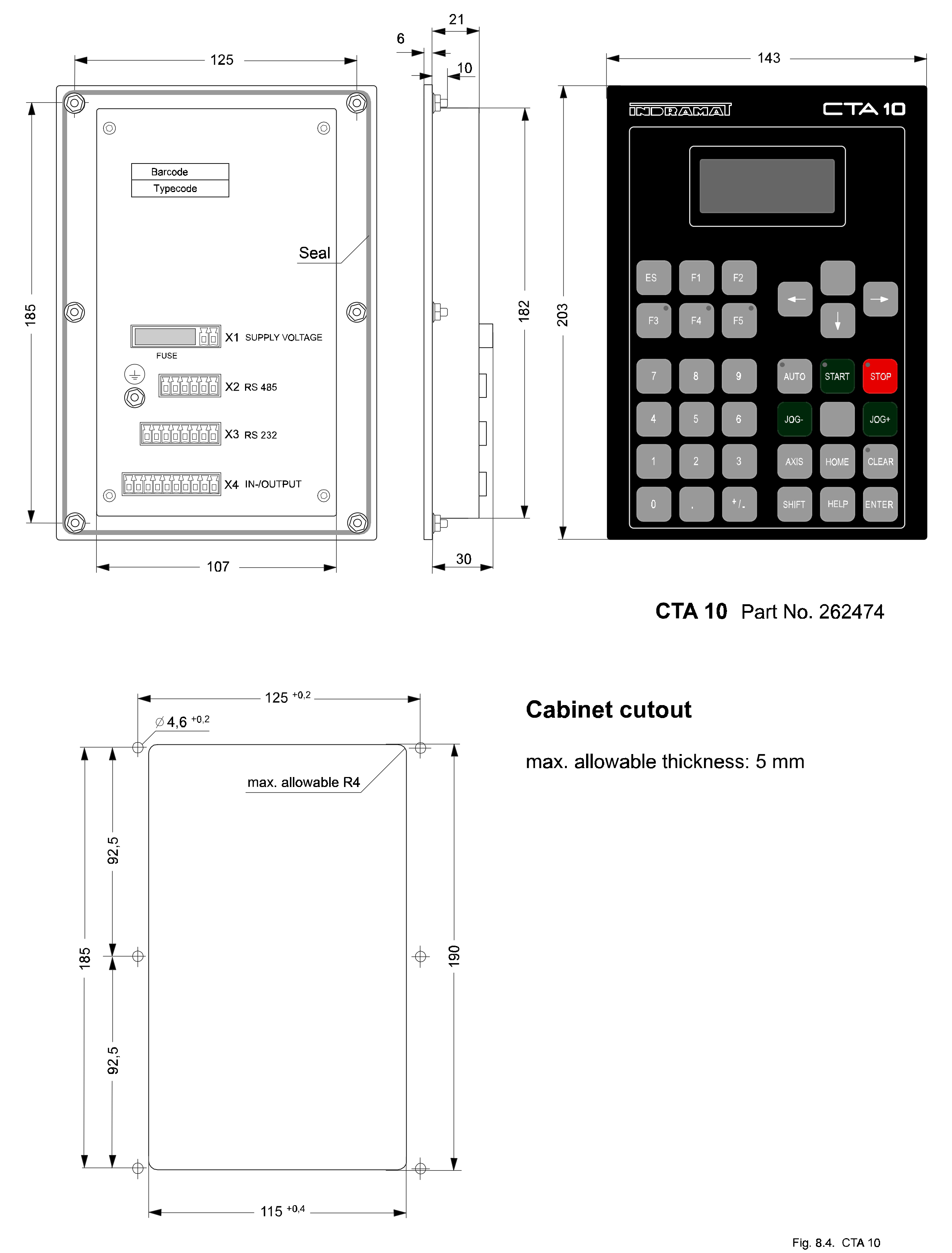


Figure D-5: CTA 10 Keyboard and Cabinet Cutout Dimensions

D.6 CTA/DLC Interconnection

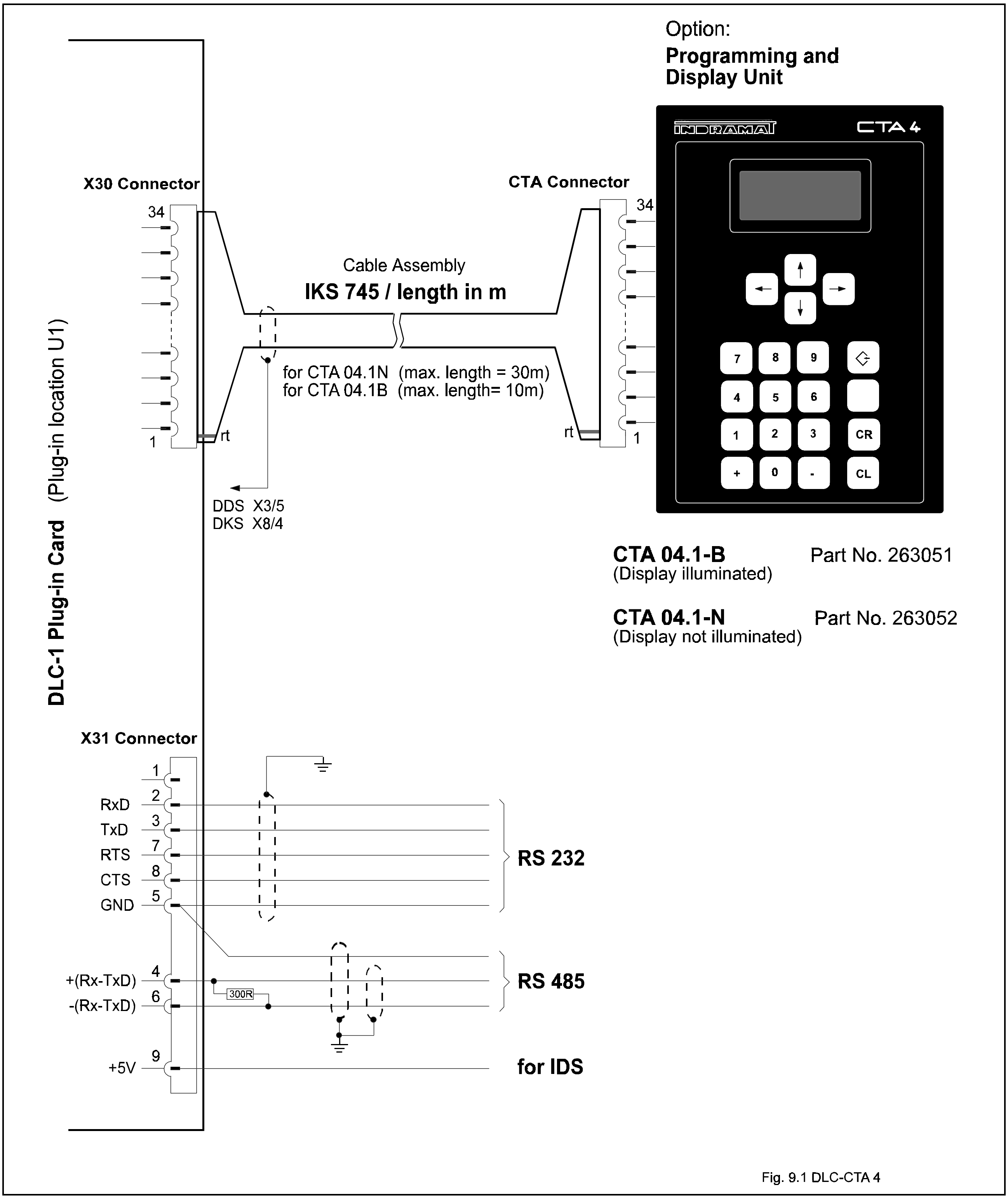


Figure D-6: CTA/DLC Interconnection

Rexroth Indramat

E DLC Type Code Descriptions

E.1 DLC Hardware Type Code Description

```
Hardware Type Code Designation:        DLC - X.X
                                       ---   ---
                                        |     |
Indramat Position Control Designation ___|     |
                                              |
DLC Hardware Version Number  _________________|
    1.1 Original Version
    2.1 With I/O bus card connector
```

E.2 Software Type Code Description

```
Software Type Code Designation:            D X - XX.X  -  0X.X
                                           - -   ----     ----
                                           | |    |        | |
Indramat Position Control Designation ______| |    |        | |
                                             |    |        | |
Software Type ________________________________|    |        | |
   A = Standard                                   |        | |
   R = Feed to Length                             |        | |
   G = G-Code Programming                         |        | |
                                                  |        | |
Corresponding DLC Hardware Version Number _________|        | |
   01.1 Original Version                                   | |
                                                           | |
Software Revision Number __________________________________| |
   (Supported with Documentation)                            |
                                                             |
Software Minor Revision Number ______________________________|
   (Not Supported with Documentation)
```

E.3 IDS Hardware Type Code Description

```
Hardware Type Code Designation:             IDS - X . X - X
                                            ---   -- --  --
                                             |    |   |   |
Product Type _________________________________|    |   |   |
                                                  |   |   |
Ids Hardware Version Number ______________________|   |   |
                                                      |   |
IDS Hardware Revision Number _________________________|   |
                                                          |
Number of Digits to the Right of  ________________________|
  the Decimal Point (0 to 3)
  (These decade switches will be in Red as opposed to Black.)
```

E.4 IDS Software Type Code Description

```
Software Type Code Designation:              IDS - X . X - X
                                             ---   -- --  --
                                              |    |   |   |
Product Type _________________________________|    |   |   |
                                                   |   |   |
IDS Software Version Number _______________________|   |   |
                                                       |   |
IDS Software Revision Number __________________________|   |
                                                           |
Corresponding DLC Software Version (Letter) _______________|
  Required For The IDS Software To Work
```

E.5 SOT Hardware Type Codes

```
Hardware Type Code Designation:               SOT XX X X X - XX
                                              --- -- - - -   --
                                               |   | | | |    |
Model Name: SOT Station Operator Terminal _____|   | | | |    |
                                                   | | | |    |
Hardware Version ______02= 16 line display_________| | | |    |
                                                     | | |    |
Construction ________________________________________| | |    |
   E = Station Unit                                    | |    |
                                                       | |    |
RAM Capacity __________________________________________| |    |
   2 = 512 KB                                            |    |
                                                         |    |
Keyboard ________________________________________________|    |
   A = Standard                                               |
   R = Rollfeed                                               |
                                                              |
DLC Application ______________________________________________|
   CU = Standard (Programmable with ScreenManager™)
   CR = Rollfeed
```

E.6 SOT Software Type Codes

```
Software Type Code Designation:               S C 2.00 - XX.02.X X
                                              - - - --   -- -- - -
                                              | | | |    |  |  | |
Model Name: SOT Station Operator Terminal ____| | | |    |  |  | |
                                                | | |    |  |  | |
Control Type ___________________________________| | |    |  |  | |
   C = DLC                                        | |    |  |  | |
                                                  | |    |  |  | |
Hardware Version Number __________________________| |    |  |  | |
                                                    |    |  |  | |
Revision Number ____________________________________|    |  |  | |
                                                         |  |  | |
Software Version ________________________________________|  |  | |
   CU = Standard (Programmable with ScreenManager)          |  | |
   CR = Rollfeed                                            |  | |
                                                            |  | |
Software Revision __________________________________________|  | |
                                                               | |
Revision Supported with Documentation _________________________| |
                                                                 |
|Minor Revision (Not Supported with Documentation)_______________|
```

I Index

Rexroth Indramat

L

M

O

P

R

S

T

Customer Service Locations

Americas (United States, Canada, and Latin America)

USA	USA	INDRAMAT Service Hotlines	Canada
Rexroth Corporation INDRAMAT Division 5150 Prairie Stone Parkway Hoffman Estates, Illinois 60192 Phone: 847/645-36 00 Fax: 857/645-62 01	Rexroth Corporation INDRAMAT Division 2110 Austin Avenue Rochester Hills, Michigan 48309 Phone: 810/853-82 90 Fax: 810/853-82 90	USA: 1-800-860-1055 Canada: 1-847-645-3600 (request Service Department) Latin America: (US Phone No.) 847-645-3600 (request Service Department)	Basic Technologies Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Phone: 905/335-55 11 Fax: 905/335-41 84
Mexico	**Argentina**	**Argentina**	**Brazil**
Motorización y Diseño de Controles, S.A. de C.V. Av. Dr. Gustavo Baz No. 288 Col. Parque Industrial la loma Apartado Postal No. 318 54060 Tlalnepantla Estado de Mexico Mexico Phone: 5/397 86 44 Fax: 5/398 98 88	Mannesmann Rexroth S.A.I.C. Division INDRAMAT Acassusso 48 41/7 1605 Munro (Buenos Aires) Argentina Phone: 01/756 01 40 01/756 02 40 Telex: 262 66 rexro ar Fax: 01/756 01 36	Nakase Asesoramiento Tecnico Diaz Velez 2929 1636 Olivos (Provincia de Buenos Aires) Argentina Phone: 01/790 52 30	Mannesmann Rexroth Automação Ltda. Divisão INDRAMAT Rua Georg Rexroth, 609 Vila Padre Anchieta BR-09.951-250 Diadema-SP Caixa Postal 377 BR-09.901-970 Diadema-SP Phone: 011/745 90 65 011/745 90 70 Fax: 011/745 90 50

Customer Service in the Americas

Asia, Australia, and Far East

Australia	China	China	China
Australian Industrial Machinery Services Pty. Ltd. Unit 3/45 Horne ST Campbellfield VIC 2061 Australia Phone: 03/93 59 0228 Fax: 03/93 59 02886	Rexroth (China) Ltd. Shanghai Office Room 206 Shanghai Intern. Trade Centre 2200 Yanan Xi Lu Shanghai 200335 P.R. China Phone: 021/627 55 333 Fax: 021/627 55 666	Rexroth (China) Ltd. Shanghai Parts & Service Centre 199 Wu Cao Road, Hua Cao Minhang District Shanghai 201 103 P.R. China Phone: 021/622 00 058 Fax: 021/622 00 068	Rexroth (China) Ltd. 1430 China World Trade Centre 1, Jianguomenwai Avenue Beijing 100004 P.R. China Phone: 010/50 50 380 Fax: 010/50 50 379
China	**Hong Kong**	**India**	**Japan**
Rexroth (China) Ltd. A-5F., 123 Lian Shan Street Sha He Kou District Dalian 116 023 P.R. China Phone: 0411/46 78 930 Fax: 0411/46 78 932	Rexroth (China) Ltd. 19 Cheung Shun Street 1st Floor, Cheung Sha Wan, Kowloon, Hong Kong Phone: 741 13 51/-54 or 741 14 30 Telex: 3346 17 GL REX HX Fax: 786 40 19 786 07 33	Mannesmann Rexroth (India) Ltd. INDRAMAT Division Plot. 96, Phase III Peenya Industrial Area Bangalore - 560058 India Phone: 80/839 21 01 80/839 73 74 Telex: 845 5028 RexB Fax: 80/839 43 45	Rexroth Co., Ltd. INDRAMAT Division I.R. Building Nakamachidai 4-26-44 Tsuzuki-ku, Yokohama 226 Japan Phone: 045/942-72 10 Fax: 045/942-03 41
Korea	**Korea**		
Rexroth-Seki Co Ltd. 1500-12 Da-Dae-Dong Saha-Gu, Pusan, 604-050 Korea Phone: 051/264 90 01 Fax: 051/264 90 10	Seo Chang Corporation Ltd. Room 903, Jeail Building 44-35 Yoido-Dong Youngdeungpo-Ku Seoul, Korea Phone: 02/780-82 07 ~9 Fax: 02/784-54 08		

Customer Service in Asia, Australia, and the Far East

Europe

Austria	Austria	Belgium	Denmark
G.L.Rexroth Ges.m.b.H. Geschäftsbereich INDRAMAT Hägelingasse 3 A-1140 Wien Phone: 1/9852540-400 Fax: 1/9852540-93	G.L.Rexroth Ges.m.b.H. Geschäftsbereich INDRAMAT Randlstraße 14 A-4061 Pasching Phone: 07229/4401-36 Fax: 07229/4401-80	Mannesmann Rexroth N.V.-S.A. INDRAMAT Division Industrielaan 8 B-1740 Ternat Phone: 02/5823180 Fax: 02/5824310	BEC Elektronik AS Zinkvej 6 DK-8900 Randers Phone: 086/447866 Fax: 086/447160
England	**Finland**	**France**	**France**
Mannesmann Rexroth Ltd. INDRAMAT Division 4 Esland Place, Love Lane Cirencester, Glos GL7 1YG Phone: 01285/658671 Fax: 01285/654991	Rexroth Mecman OY Riihimiehentie 3 SF-01720 Vantaa Phone: 0/848511 Fax: 0/846387	Rexroth - Sigma S.A. Division INDRAMAT Parc des Barbanniers 4, Place du Village F-92632 Gennevilliers Cedex Phone: 1/41475430 Fax: 1/47946941	Rexroth - Sigma S.A. Division INDRAMAT 91, Bd 1 Joliot Curie F-69634 Venissieux - Cx Phone: 78785256 Fax: 78785231
France	**Italy**	**Italy**	**Netherlands**
Rexroth - Sigma S.A. Division INDRAMAT 270, Avenue de lardenne F-31100 Toulouse Phone: 61499519 Fax: 61310041	Rexroth S.p.A. Divisione INDRAMAT Via G. Di Vittoria, 1 I-20063 Cernusco S/N.MI Phone: 02/92365-270 Fax: 02/92108069	Rexroth S.p.A. Divisione INDRAMAT Via Borgomanero, 11 I-10145 Torino Phone: 011/7712230 Fax: 011/7710190	Hydraudyne Hydrauliek B.V. Kruisbroeksestraat 1a P.O. Box 32 NL-5280 AA Boxtel Phone: 04116/51951 Fax: 04116/51483
Spain	**Spain**	**Sweden**	**Switzerland**
Rexroth S.A. Centro Industrial Santiago Obradors s/n E-08130 Santa Perpetua de Mogoda (Barcelona) Phone: 03/718 68 51 Telex: 591 81 Fax: 03/718 98 62	Goimendi S.A. División Indramat Jolastokieta (Herrera) Apartado 11 37 E-San Sebastion, 20017 Phone: 043/40 01 63 Telex: 361 72 Fax: 043/39 93 95	AB Rexroth Mecman INDRAMAT Division Varuvägen 7 S-125 81 Stockholm Phone: 08/727 92 00 Fax: 08/64 73 277	Rexroth SA Département INDRAMAT Chemin de l`Ecole 6 CH-1036 Sullens Phone: 021/731 43 77 Fax: 021/731 46 78
Switzerland	**Russia**		
Rexroth AG Geeschäftsbereich INDRAMAT Gewerbestraße 3 CH-8500 Frauenfeld Phone: 052/720 21 00 Fax: 052/720 21 11	Tschudnenko E.B. Arsenia 22 153000 Ivanovo Russia Phone: 093/22 39 633		

Customer Service in Europe

Germany

Central Sales Region	Eastern Sales Region	Western Sales Region	Northern Sales Region
INDRAMAT GmbH Bgm.-Dr.-Nebel-Str. 2 D-97816 Lohr am Main Phone: 09352/40-0 Fax: 09352/40-4885	INDRAMAT GmbH Beckerstraße 31 D-09120 Chemnitz Phone: 0371/3555-0 Fax: 0371/3555-230	INDRAMAT GmbH Hansastraße 25 D-40849 Ratingen Phone: 02102/4318-0 Fax: 02102/41315	INDRAMAT GmbH Fährhausstraße 11 D-22085 Hamburg Phone: 040/227126-16 Fax: 040/227126-15
Southern Sales Region	**Southwestern Sales Region**	**INDRAMAT Service Hotline**	
INDRAMAT GmbH Ridlerstraße 75 D-80339 München Phone: 089/540138-30 Fax: 089/540138-10	INDRAMAT GmbH Böblinger Straße 25 D-71229 Leonberg Phone: 07152/972-6 Fax: 07152/972-727	INDRAMAT GmbH Phone: 0172/660 040 6 -or- Phone: 0171/333 882 6	

Customer Service in Germany

286834

Printed in Germany

Rexroth Indramat